



**VOICE AND VIDEO CAPACITY OF A  
SECURE WIRELESS SYSTEM**

THESIS

Jason R. Seyba, 1<sup>st</sup> Lieutenant, USAF  
AFIT/GCS/ENG/07-14

**DEPARTMENT OF THE AIR FORCE  
AIR UNIVERSITY**

***AIR FORCE INSTITUTE OF TECHNOLOGY***

---

**Wright-Patterson Air Force Base, Ohio**

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GCS/ENG/07-14

VOICE AND VIDEO CAPACITY OF A  
SECURE WIRELESS SYSTEM

THESIS

Presented to the Faculty  
Department of Electrical and Computer Engineering  
Graduate School of Engineering and Management  
Air Force Institute of Technology  
Air University  
Air Education and Training Command  
in Partial Fulfillment of the Requirements for the  
Degree of Master of Science

Jason R. Seyba, BS  
1<sup>st</sup> Lieutenant, USAF


June 2007

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

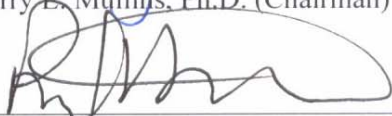
VOICE AND VIDEO CAPACITY OF A  
SECURE WIRELESS SYSTEM

Jason R. Seyba, BS  
First Lieutenant, USAF

Approved:

  
\_\_\_\_\_  
Barry E. Mullins, Ph.D. (Chairman)

30 May 07  
Date

  
\_\_\_\_\_  
Paul D. Williams, Ph.D. (Member)

30 May 07  
Date

  
\_\_\_\_\_  
Richard A. Raines, Ph.D. (Member)

30 May 07  
Date

## **Abstract**

This thesis investigates techniques for improving the security and availability of secure wireless multimedia systems. Three effects are discussed in this study: 1) The effect on the audio-video capacity of the wireless system by securing the voice signal is measured; 2) The effect of the audio-video traffic originating foreign to the measured wireless local area network, thus using an access point to forward the traffic is evaluated; and 3) The effect of using an audio-only signal compared with an audio-video signal is also related to the capacity of the wireless system. The effects are determined experimentally using 36 human subjects interacting with a wireless multimedia system which was developed as part of this thesis effort. Additionally, techniques for deploying wireless multimedia systems including the maturity and security of the technology are addressed. Analysis of the experimental data suggests that securing the voice traffic has no significant effect on the quality of the audio signal received, which indicates that the system has a good design. Additional analysis of the data suggests that an 18.4% improvement in the perceived quality of the audio signal can be made by routing the audio and video traffic through an access point instead of allowing the audio and video traffic to flow directly between two arbitrary nodes within a wireless local area network. Results suggest that increasing the number of conversations reduces the perceived quality of the audio signal by 23.5% and the video signal by 16.8%. Additionally, results suggest that by disabling video capability, the perceived quality of audio increases by 38.9%. Existing analytical and computer models estimate the audio and video capacity at eleven conversations. The empirical data within this thesis suggests that this an overestimation of audio-video capacity which is caused by assuming bit error rates of 0.1 bits/sec accurately model wireless networks. This study suggests that the true capacity of a wireless multimedia system using 802.11g is one audio-only conversation on the wireless network.

AFIT/GCS/ENG/07-14

*To Father and Mother*

## **Acknowledgments**

I would like to express my sincere appreciation to my faculty advisor, Dr. Barry E. Mullins, for his guidance and support throughout the course of this thesis effort. The insight and experience was certainly appreciated. I would, also, like to thank my sponsors, Mr. Edward E. Boyle, Lt Col Gregory L. Bonafede, Paul D. Faas and Joseph E. Lyons, all from the Logistics Readiness Branch of the Air Force Research Laboratory for the support and latitude provided to me in this endeavor.

Jason R. Seyba

## Table of Contents

	Page
Abstract .....	iii
Acknowledgements .....	v
Table of Contents .....	vi
List of Figures .....	viii
List of Tables .....	x
I. Introduction .....	1
1.1 Background .....	1
1.2 Problem Definition .....	1
1.3 Goals.....	2
1.4 Approach .....	2
1.5 Preview .....	3
1.6 Implications .....	3
1.7 Summary .....	4
II. Literature Review .....	5
2.1 Session Initialization Protocol.....	5
2.2 Real Time Protocol.....	12
2.3 Audio Codecs .....	14
2.4 Wireless Quality of Service Options .....	16
2.5 Key Distribution Methods .....	17
2.6 Authentication, Integrity and Replay Protection .....	21
2.7 Advanced Encryption Standard.....	22
2.8 Block Cipher Modes of Operation for Secure Real-Time Transport Protocol .....	27
2.9 Methods for obtaining MOS (Mean Opinion Score).....	28
2.10 Related Research .....	29
2.10 Conclusion.....	32
III. Methodology .....	33
3.1 Experimental Design .....	34
3.2 Architecture of Experimental Software .....	42
3.3 Wireless Network Setup .....	42
3.4 Conclusion .....	43



	Page
IV. Results and Analysis .....	44
4.1 Analysis of Significance.....	44
4.2 Audio MOS Analysis .....	45
4.2.1 Analysis of All Audio MOS Data .....	45
4.2.2 Analysis of the With- and Without-AP Audio MOS Data .....	48
4.2.3 Analysis of the One Conversation Audio MOS Data.....	49
4.2.4 Analysis of the Two Conversation Audio MOS Data .....	53
4.2.5 Analysis of the Audio-Only Audio MOS Data.....	54
4.2.6 Analysis of Video-Enabled Audio MOS Data .....	56
4.3 Analysis of Video MOS Data.....	57
4.4 Analysis of Intelligibility Data .....	60
4.5 Conclusion.....	61
V. Conclusion.....	62
5.1 Summary of Analysis .....	62
5.1.1 Study Questions and Answers .....	62
5.1.2 Secondary Data Effects .....	63
5.2 Recommendations for Future Research.....	64
5.3 Relevance of the Current Investigation .....	64
5.4 Conclusion.....	65
Appendix A. Secure Wireless Multimedia Information Consent Letter .....	66
Appendix B. Secure Wireless Multimedia Informed Sheet .....	68
Appendix C. Secure Wireless Multimedia Raw Data .....	70
Appendix D. Additional Subject Data Analysis.....	73
Appendix E. Software Architecture .....	80
Bibliography .....	98
Vita .....	101

## List of Figures

Figure	Page
1. SIP Overview .....	6
2. H.323 and SIP Architectures .....	8
3. Scaleable SIP Session Initialization .....	9
4. Registration with Location Server.....	10
5. Session Description Protocol Example .....	11
6. RTP Packet Structure .....	12
7. RTP Header Structure.....	14
8. CELP Synthesis Filter .....	16
9. Diffie-Hellman Key Distribution .....	18
10. Man in the Middle Attack on Diffie-Hellman.....	19
11. Zimmerman's Solution to Diffie-Hellman Vulnerability .....	20
12. Building a 256-bit Key in four steps using the AES Key Schedule .....	23
13. AES Core Procedure for Key Generation.....	24
14. AES Encryption Step One .....	25
15. AES Encryption Step Two .....	26
16. AES Encryption Step Three .....	26
17. AES Encryption Step Four .....	26
18. Counter (CTR) Mode Encryption.....	27
19. Output Feedback (OFB) Mode Encryption .....	28
20. Network Conditions .....	36
21. Software User Interface for Audio/Video MOS.....	38
22. Software User Interface for Audio Intelligibility .....	41

Figure	Page
35. 2-way Effect Between the Number of Conversations and Use of Video.....	46
36. 2-way Effect Between the Number of Conversations and Use of Video (Unsecured).....	47
37. 2-way Effect Between the Number of Conversations and Use of Video (Secured) .....	48
38. 2-way Effect Between the Use of Video and Security During One Conversation .....	50
39. 2-way Effect Between Use of Video and Security (One Conversation with AP).....	51
40. 2-way Effect Between the Number of Conversations by Security With Audio-Only .....	54

## List of Tables

Table	Page
1. Rijndael S-Box .....	23
2. MOS Quality and Impairment Numerical Relationship .....	28
3. MOS Scores of Various Codecs .....	29
4. Recommended Values for $T_{\text{BACKOFF}}$ Calculation .....	30
5. Recommended Values for $T_{\text{TX}}$ Calculation .....	31
6. Analytical Model of Audio-Video Capacity .....	31
7. Network Conditions .....	35
8. Subject Treatments .....	40

# VOICE AND VIDEO CAPACITY OF A SECURE WIRELESS SYSTEM

## **I. Introduction**

### **1.1 Background**

Decisions made now regarding implementation strategies for collaborative applications that rely upon multimedia capability will result in production systems that will inherit the security and voice quality features provided by the underlying communication protocols. The Air Force has shown increased interest in collaborative applications that allow multiple users to conduct voice and video communication and other multimedia services using applications built upon the IP (Internet Protocol) and the RTP (Real Time Protocol). These efforts may provide a cost savings and improvement in the effectiveness of logistics operations in applications such as aerial port operations, tanker mission support planning, as well as the coordination of flight line maintenance activities. One of the technology gaps that exist within the realm of secure wireless multimedia is the ability to integrate secure voice and video features into existing applications. This thesis provides the technical details of how to integrate secure voice and video communications into a Java application in addition to evaluating the capacity of the system to support wireless use. Additionally, the thesis provides analysis of experimental data that are used to determine the capacity of a secure audio-video wireless system.

### **1.2 Problem Definition**

Empirical data related to the capacity of secure wireless multimedia systems are often not used to verify the accuracy of analytical and computer-based simulations of this capability. This thesis provides both data and analysis of a secure wireless multimedia system's capacity and how

it relates to several analytical and computer-based simulations. Additionally, the effects of routing the traffic through an access point as well as the effect of securing the signal are investigated.

### **1.3 Goals**

This thesis tests the hypothesis that securing a voice transmission significantly reduces the voice quality in a wireless network. It can be expected that other variables to include the pixel size of the transmitted video as well as the number of simultaneous conversations is far more critical to predicting voice and video quality as opposed to the effect of adding security features to the multimedia stream. An additional question to be answered in this thesis is the effect of routing the traffic through an AP (access point) as opposed to simply having two nodes on the same WLAN (wireless local area network) exchange audio and video information. All of these questions are answered in terms of the audio and video quality as evaluated by human subjects.

### **1.4 Approach**

To accomplish the goals of this research effort, it is necessary to use human subject data in conjunction with a system that is capable of transmission and receipt of multimedia through a wireless connection. Use of IEEE 802.11g [21] (hereafter referred to as 802.11g) equipment, which is theoretically capable of 54 Mbps (megabits/sec), is used for the entire study. The software that the human subjects are exposed to must be able to enable or disable the secure features, change the number of conversations on the WLAN, enable or disable the video feature and allow routing of real-time traffic through the AP and use a node-to-node model. When the subject data are collected, the empirical results are compared with existing computer and analytical models.

## **1.5 Preview**

Empirical data collected within this study suggests that securing the voice traffic within the system does not significantly affect the quality of the audio signal received. Additionally, a significant improvement to the audio signal can be made by routing the traffic through an AP. Several internal checks for validity are conducted in the analysis section of this study. Two results suggest that increasing the number of conversations significantly decreases both audio and video quality and that disabling video significantly improves the quality of the audio connection. Overall, it was learned that a single conversation, with the video disabled, produces a 2.8 out of 5 Mean Opinion Score which is interpreted as a “fair” or “slightly annoying” audio quality. When the video capability is enabled, a 1.8 out of 5 Mean Opinion Score is obtained, which is interpreted as a “poor” or “annoying” audio quality. In the case of two audio connections, 2.2 out of 5 Mean Opinion Score is produced which also relates to a “poor” or “annoying” audio quality. More traffic than this causes a 1 out of 5 Mean Opinion Score, which is interpreted as a “bad,” “very annoying” or “unusable” audio quality. Therefore, the data suggests that no more than one audio and video conversation can occur in a WLAN with current 802.11g equipment, whereas two audio-only conversations can occur using the same equipment.

## **1.6 Implications**

The major implication of this study is that existing computer and analytical models yield different results than what was observed in this study. Specifically, computer and analytical models predicted much greater capacities than is actually possible with existing 802.11g equipment and software developed for this study. Additionally, the data within this study contains evidence that security features do not significantly affect the performance of the system.

This thesis also successfully demonstrated a method to secure real-time multimedia communication within existing Java applications.

## **1.7 Summary**

The goal of this thesis is to allow the integration of secure multimedia capability within Java applications and the evaluation of the capacity of a 802.11g WLAN to support various numbers of conversations in various modes of operation.

Chapter II describes the current state of VoIP (Voice over Internet Protocol) technology with special focus on current methods to secure real-time traffic.

Chapter III describes the thesis methodology that is used to answer the questions that is also defined in detail in the chapter. The two primary questions are: 1) Is there a significant effect on the quality of the audio and video data by routing the traffic through an AP compared with two arbitrary nodes on a WLAN transmitting and receiving audio and video traffic?; and 2) Is there a significant impact by securing the audio traffic on the quality of the audio received? Additionally, a method that was developed as part of this thesis effort to secure multimedia IP traffic when using Java applications is discussed in detail.

Chapter IV describes the results and analysis of the subject data that was collected as it relates to the questions that were developed in detail in Chapter III.

Chapter V discusses the relationship between the empirical results and existing computer and analytical models for modeling the capacity for multimedia traffic within IP networks.

Chapter VI discusses both the relevance of the current investigation and gives recommendations for future research that builds on the results of this thesis effort.



## **II. Literature Review**

This chapter describes the current state of VoIP (Voice over Internet Protocol) technology with special focus on current methods to secure the real-time traffic. Section 2.1 discusses two family of protocols that are capable of voice and video services over IP (Internet Protocol) networks: H.323 [25] and SIP (Session Initialization Protocol) [22]. Section 2.2 describes how RTP (Real Time Protocol) is used to transmit the data containing the real-time voice and video data. Section 2.3 describes how audio information is compressed so that it may be transmitted using RTP. Section 2.4 describes QoS (Quality of Service) improvements to the data link layer that are currently implemented in IEEE 802.11e standard [20]. Section 2.5 describes how to transmit keys for the purpose of encrypting the real-time traffic. Section 2.6 describes how authentication, integrity and replay protection schemes are enabled by the keyed-hash message authentication code. Section 2.7 describes how the AES (Advanced Encryption Standard) is used to encrypt the real-time content. Section 2.8 describes how the block cipher modes of operation use AES to secure a stream of real-time traffic. Section 2.9 describes how the (MOS) Mean Opinion Scores are obtained and how they relate to the quality of the audio and video traffic. Section 2.10 describes related research initiatives for determining the capacity of wireless audio-video systems. Section 2.11 summarizes the results of the literature review.

### **2.1 Session Initialization Protocol**

SIP (Session Initialization Protocol) is an application layer protocol that can use either UDP (User Datagram Protocol) or TCP (Transmission Control Protocol) to establish, modify and terminate sessions. It was initially developed by the IETF (Internet Engineering Task Force) MMUSIC (Multiparty Multimedia Session Control) committee in 1999. The RFC (Request for

Comments) that describes the protocol is RFC 3261, which has been in its current form since 2002.

Figure 1 describes how SIP can be used to setup a simple media session between two UAs (User Agents). Within both the initialization and termination packets, additional information about the session is communicated using the SDP (Session Description Protocol).

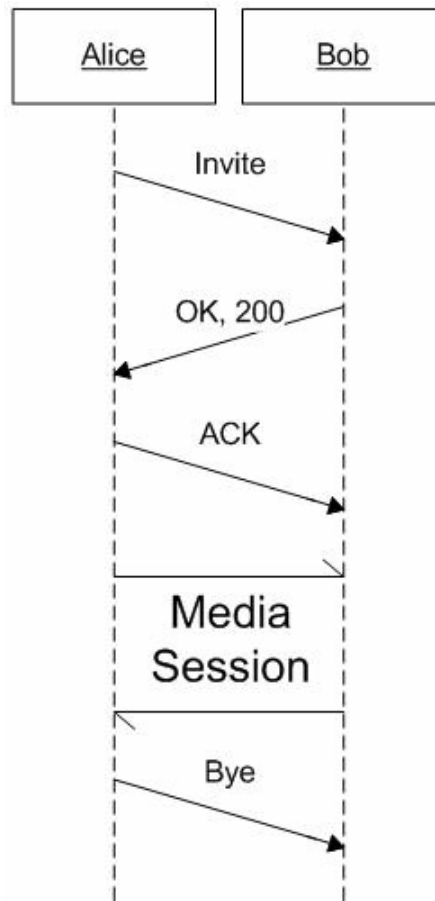


Figure 1. SIP Overview [28]

Traditional voice communication systems developed within the phone industry consist of circuit switched networks. Many of the signaling techniques used by the telephone industry were adopted by the first standard for implementing voice services over IP-based networks - H.323. Development and maintenance of this standard is coordinated by the ITU-T (International Telecommunications Union). A major design philosophy that H.323 derives from traditional

telecommunications systems is the concept that two users are connected by a “call” between handsets. This assumes that typical communication scenarios consist of two end-points, which does not easily support many users’ expectation of a teleconference capability. For this reason, implementations of the H.323 standard that provides teleconference capability generally take longer to develop, are more complicated to develop, and more difficult to maintain when compared with SIP implementations.

SIP uses a design philosophy based around the concept of a “session” which has resulted in implementations of multi-party conference capabilities. Additionally, events related to the “session” can be coordinated before the start and end of media transmissions which greatly improves user satisfaction with the final implementation. Events that could occur before or after the start of media transmission can include session description information such as the type of media to be transferred, security settings, number of users or application-specific information.

A major concern that must be addressed in regard to implementation of multimedia capabilities is how the voice components work with more traditional PSTNs (Public Switched Telephone Networks). Many gateway implementations support the integration of PSTN with SIP or H.323 standards. Additionally, hardware plug-ins for the hosting workstation are readily available. It is also possible to connect a wireless AP (Access Point) to a voice gateway workstation through a LAN (Local Area Network) to enable wireless voice capability.

Figure 2 illustrates the scope of the two families of protocols used for real-time communication – H.323 and SIP. The SIP Architecture contains a protocol named SIP which can use either TCP or UDP in the transport layer of an IP packet. Several components of the two architectures are in common such as RTP (Real-Time Protocol) packets which contain the audio and video information [35] [43]. RTP is discussed in detail in Section 2.2. An additional protocol that support SIP is RTCP (RTP Control Protocol), which is used to ensure that all endpoints have common variables related to the status and quality of the connections is common

to both architectures. Since RTP traffic must arrive in a reasonable time frame or not at all, the UDP protocol was chosen for the transport layer since it is not necessary to retransmit a packet if a loss occurs within the network. MEGACO and H.248 were developed jointly by ITU-T and IETF and are used for gateway control [34]. However, multiple streams are handled in different ways by the two architectures through two different protocols H.245 and RTSP (Real-Time Streaming Protocol).

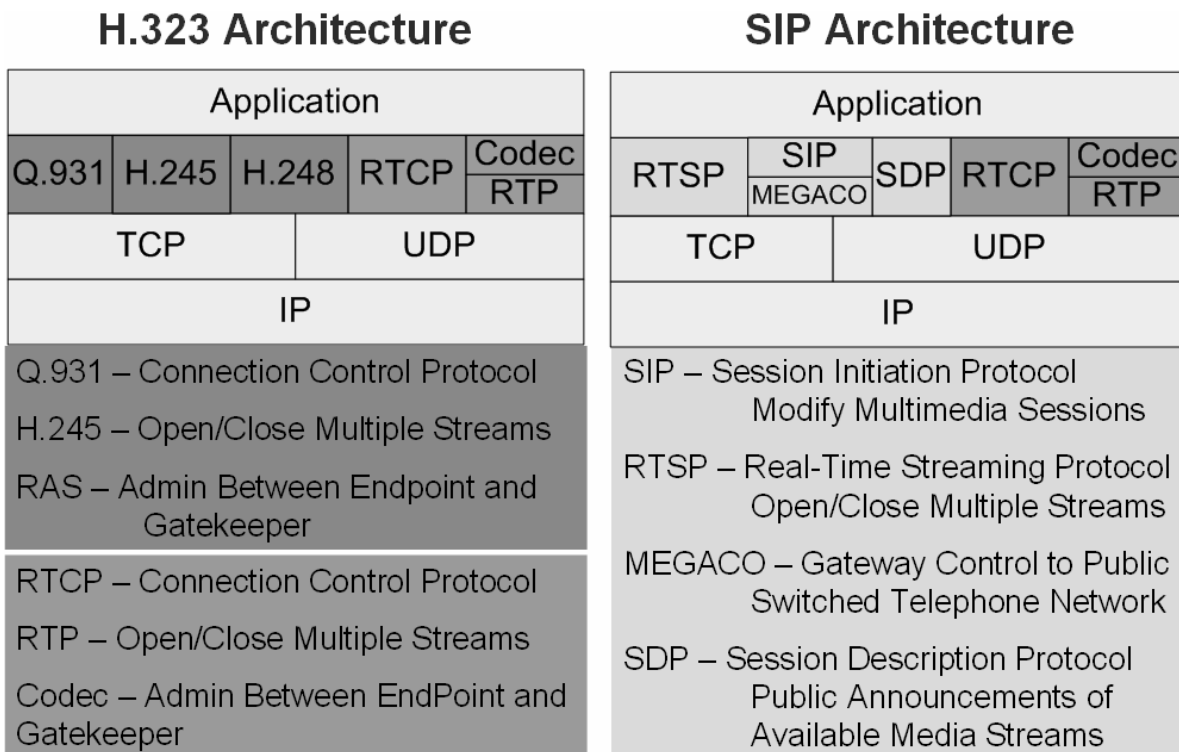


Figure 2. H.323 and SIP Architectures

Additional terminology is required to explain implementation of SIP in modern applications. The hardware device that a UA (user agent) runs on is named a SIP Terminal. Since each UA is required to be both a client and server, it is necessary to disambiguate by naming the UA that initiates a session the UAC (User Agent Client) and the UA that acts as the responder the UAS (User Agent Server). Modern implementations have proxy servers that each

UA communicates with. An outgoing proxy is associated with the UAC, and an incoming proxy is associated with a UAS. Additionally, the server may or may not read the content of the SDP (Session Description Protocol). SDP describes the characteristics of the session, and its details are explained at the end of this section. If SDP state is maintained by the proxy, it is called a stateful proxy server, and if it is not retained, the proxy is a stateless proxy server.

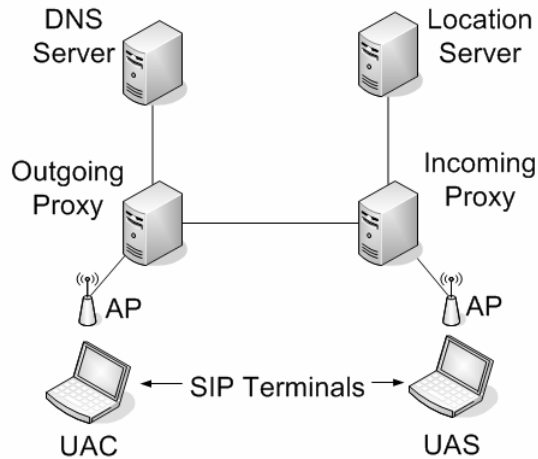


Figure 3. Scalable SIP Session Initialization [25]

Figure 3 displays a possible infrastructure that can support a scalable SIP wireless solution. It is possible that both the UAC and UAS are wireless devices that communicate through an AP that is connected to a proxy server. In order to do a lookup for the inbound proxy server, it is necessary for a DNS Server be associated with the outbound proxy server. An INVITE message is sent first to the outgoing proxy server. Next, the INVITE is sent to the incoming proxy server with the assistance of an IP address supplied by the DNS server. After the location server responds with the current address of the UAS, the inbound proxy server forwards the INVITE message. Three messages are returned to the UAC in this scenario: the inbound proxy server responds with a 100 Trying message after receiving the INVITE; the UAS responds with a 180 Ringing message after receiving the INVITE; and a 200 OK is transmitted after the multimedia session has been accepted by the user operating the UAS.

A registrar server must be present to allow the UA to update their current IP address to be stored by the location server. Updating a location server consists of only two SIP commands. The REGISTER command is sent from the UA to the registrar server, and a 200 OK is replied once the registrar server has successfully updated the location server. The Update and Response messages are typically proprietary protocols used by hardware manufacturers such as Cisco.

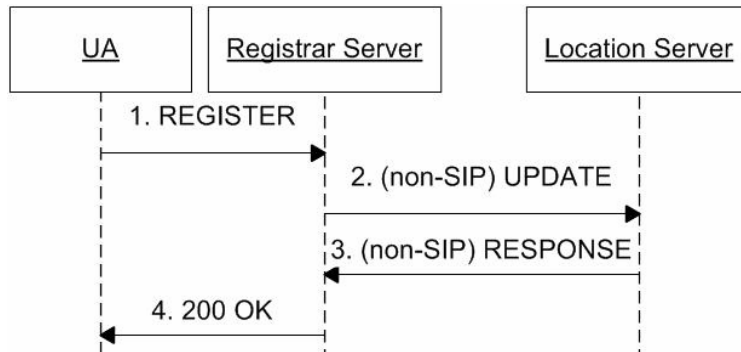


Figure 4. Registration with Location Server [25]

There exists many legitimate security concerns regarding the use of location servers. For this reason, it is important that the location server verify the identity of all users attempting to retrieve location information. Additionally, robust policies must be in place to map who is allowed access to the location of users who have accounts on the location server. Additionally, it may be necessary to add time and location rules in addition to generic data access rights. The advantages of a location server include that a particular user maintains a single SIP URI (Universal Resource Identifier), which contains the essential information to attempt the initiation of a session with the associated user. A rough analogy is that a URI is to multimedia sessions as email address is to email. Additionally, the use of a location server is a scalable solution since each UAC that corresponds with the UAS does not need to be updated.

Another protocol that was briefly discussed in Figure 2 is the Real Time Streaming Protocol (RTSP) which serves the purpose of opening and closing a media stream. This

capability is similar to a HTTP (Hypertext Transfer Protocol) session, however, the RTSP protocol is more flexible in that it allows the UAS or UAC to initiate requests for media streams opening or closing. Additionally, HTTP does not maintain the state of the connection that it manages. RTSP maintains state during the entire period the media stream is active, allowing parameters to be maintained without rebuilding the connection.

As stated before, SIP relies on the SDP for a large portion of the content passed in SIP messages [3]. Supported session description settings include lip synchronization (LS), flow identification, and specific audio/video codecs. Since it is common that multiple streams exist within one session, it is desirable to distinguish between group attributes and stream attributes. An example of how SDP can be used to initialize a session is in Figure 5. Line 1 states that for all streams in the group, lip synchronization should occur. Line 2 states that the primary stream is the audio channel on port 30000; the “RTP/AVP 0” value indicates that the RTP (Real Time Protocol) should transport Attribute Value Pair (AVP) 0. When a lookup is conducted on AVP 0, it is discovered that it is referring to G.711 audio codec [23]. Lines 3 and 5 define the media identification values (mid) which are defined in lines 4 and 6 respectively. Line 4 describes the video codec (H.261) and port number (30002) to be used in a manner similar to Line 2.

1	a=group:LS 1 2
2	m=audio 30000 RTP/AVP 0
3	a=mid:1
4	m=video 30002 RTP/AVP 31
5	a=mid:2
6	m=audio 30004 RTP/AVP 0

Figure 5. Session Description Protocol Example [29]

## 2.2 Real Time Protocol

The actual transmission of the voice data is conducted within RTP (Real Time Protocol) packets. For this reason, to understand the performance and security of the voice stream, a thorough investigation must be made of the structure of the RTP packet. Figure 6 illustrates that 20 bytes are used in the header for the IPv4 header. Since the packet uses the UDP because of the real-time nature of the broadcast, 8 bytes are added for the UDP header. Finally, the RTP header consists of a minimum of 12 bytes. Also, the codec block size varies greatly depending on the exact audio/video codec used.

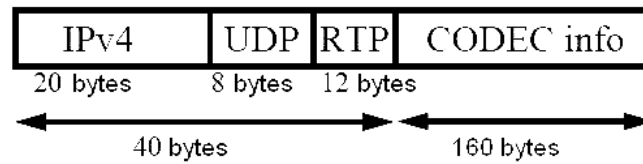


Figure 6. RTP Packet Structure [29]

RTP packets are generally sent every  $20\mu\text{s}$ . This amount of time was chosen because a human is barely able to detect the absence of  $20\mu\text{s}$  of audio content. Unacceptable network congestion can also be caused by sending too many RTP packets. For this reason, it is possible that the codec data may actually be fewer bytes in size than the header. For instance, if the G.729 audio codec [24] is used instead, the size of the codec information that exists within one RTP packet is 20 bytes, or half the size of the header.

There are many considerations for real-time application data that need not be considered in typical TCP traffic. Among two of the most important aspects that must be considered is the order and time in which an individual packet arrives. For this reason, it is necessary that the RTP packet contain both timestamp and sequence number. These two attributes also ensure that the UAS receiving the real-time traffic knows the difference between lost packets and time-periods in which the UAC did not send data because the user was not speaking. It should be noted that



when there exists either silence because a packet has been delayed in the network, most implementations play white noise instead of silence for sound quality reasons.

One of the most serious problems with highly loaded networks supporting real-time data is variations in delay. This phenomenon is called jitter. Humans can detect a wobbling sound in the audio stream if jitter is not handled by the UAS. It is now common practice to implement a jitter buffer that only sends the audio data to hardware once the timer associated with the buffer has expended. An additional concern is that of clock synchronization between the two SIP applications. The total time to send the voice data to the receiver can thus be characterized as the sum of audio collection time, local clock synchronization time, UAS packaging delay, network delay, and UAC jitter-buffer delay. An additional constraint on the real-time multimedia application is that it must ensure that the total delay incurred must be less than 177ms, as above this range, humans begin to notice that a delay is occurring in the audio channel [17].

The exact composition of the RTP header as described in RFC 1889 is shown in Figure 7. Figure 6 illustrated 12 bytes of mandatory RTP header content; however Figure 7 illustrates an extra 32-bit word. The extra 32-bit word is used identify extra contributing sources beyond the synchronization source. The header of the RTP packet shown in Figure 7 consists of a version number (VER), a flag to indicate whether padding exists (P), a flag to indicate that this packet is an extension (X), the number of contributing sources to the stream (CC – Contributing Source Count), a marker to indicate a major change in the data stream ea. a period of talking after a period of silence (M), a type of payload that was defined in SDP in the session initialization (PTYPE), a sequence number for the packet, a timestamp, and a list of all IP addresses that contributed to the content of the packet.

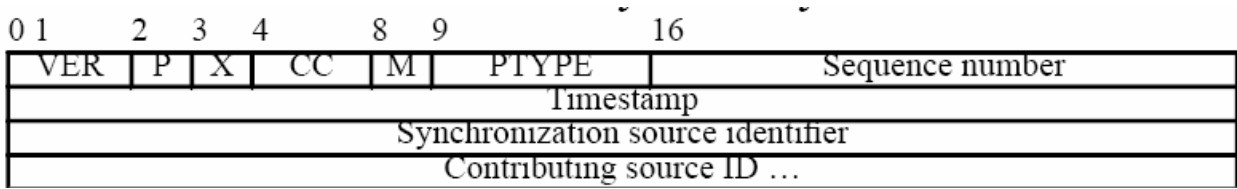


Figure 7. RTP Header Structure [29]

Another protocol that is often implemented in real-time multimedia applications is the RTP Control Protocol (RTCP). The purpose of this protocol is to ensure that both UAs have a common understanding of the network conditions. Examples of how this information may be used by the application include modifying the codec or adjusting the rate of sent into the network. Mandatory fields within RTCP include a version number, a flag for the existence of padding, a report count that must equal or exceed two, a type of payload, the length of all reports, followed by the actual payload. The actual payload is made up of a series of reports, most typically both a receiver and sender report are sent by both UAs. These reports contain the information needed for each UA to communicate the loss rate, discard rate, burst duration, gap duration, round trip delay, end system delay, noise level, and jitter buffer size that the other UA need to control the real-time stream.

### 2.3 Audio Codecs

Packaging the voice data within the RTP packets can be described by using two specific audio codecs. G.711 is a frequently used codec because it does not significantly compress the audio which results in high voice quality, but with the drawback of a high bandwidth requirement [20]. G.729 allows a compression of the voice data resulting in a lower quality than G.711, however, it is available at 1/3 of the bandwidth of a G.711 codec [24].

The technology enabling G.711 is Pulse Code Modulation (PCM). Since humans measure the tone of sound logarithmically, PCM compresses sound logarithmically. It allows a

of the human. The law typically used in implementations of PCM within the United States is known as the mu-Law [19] and is:

$$y = \frac{\ln(1 + ux)}{\ln(1 + u)} \quad (1)$$

where  $u = 255$  and the value to compress is set as the value  $x$ , the value  $y$  is the resulting value.

G.729 uses CELP (Code Excited Linear Prediction) to compress the data which takes advantage of the fact that human speech can be modeled as a series of shapes of frequency or “formants” at a particular pitch [24]. By matching a particular excitation vector to the formant of the word that was spoken, and by matching the pitch to one of eight possible values, a significant compression is possible with little loss of audio content. A total of 10 bytes is all that is encapsulated in the RTP packet. The 10 bytes contain a representation of one of the 128 excitation vectors along with one of eight pitch frequencies.

A pitch synthesis filter is used to determine the correct combination of pitch and excitation vectors. Several variables need to be defined before the CELP pitch synthesis filter in Figure 8 can be described. The pitch of the signal is “G” and the excitation vector is “ $x^{(i)}(n)$ ”. Together, they are known as the scaled codeword and compared with the input speech  $s(n)$ . Two filters to determine the pitch and formant are referred to as  $P(z)$  and  $H(z)$  respectively. A weighting filter uses a constant  $\gamma = 1.25$  for ensuring that the gain is not related to the error between the input speech and the scaled codeword.

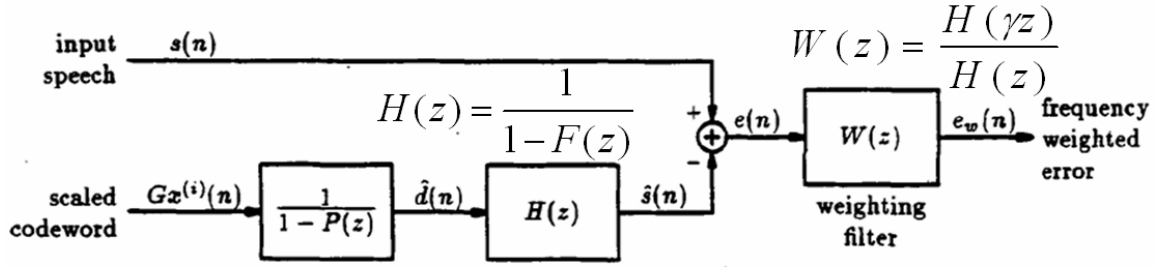


Figure 8. CELP Pitch Synthesis Filter [42]

## 2.4 Wireless Quality of Service Options

The final quality of the signal that is experienced by the end user is affected by not only the type of codec, but also by the quality of the network connection. Due to the high sensitivity of real-time multimedia to delays and packet losses within the network, the data link layer is highly important as any delays resulting from inefficiencies in terms of packet losses and packet delays cannot be made up in transport or application layer changes. For this reason, much effort has been expended to ensure that the IEEE 802.11 family of protocols provides a high quality of service (QoS). For example, use of the PCF (Point Coordination Function) protocol allows a contention-free period to allow the access point (AP) to poll each station to determine how the rest of the contention free period is used. Since the PCF has numerous limitations, HCF (Hybrid Coordination Function) was developed under the IEEE 802.11e standard [20]. Enhanced Distributed Coordination Function (EDCA) is one of the two modes under which HCF can function to provide QoS. EDCA allows the distinction of classes of traffic, in addition to providing a transmission opportunity (TXOP) for each station to send as many frames as possible. HCF Controlled Channel Access (HCCA) is the second HCF mode that has the benefits of both EDCA and PCF combined to make it the most advanced protocol, allowing customization of the quality of each of the streams within the WLAN. However, only the EDCA mode is mandatory in the 802.11e standard.

## 2.5 Key Distribution Methods

The issue of security of the multimedia traffic is a highly important variable to consider when deploying real-time multimedia capabilities. The currently accepted method of securing multimedia traffic is defined in RFC 3711 [36]. The Secure Real-Time Transport Protocol (SRTP) was defined in March of 2004 to secure the RTP against multiple attacks on the application layer [15]. Specific defenses that SRTP provides include encryption of the data stream, authentication, and protection of both the integrity and replay ability of the message. Additionally, SRTP provides for both the unicast and multicast modes so that no capability of RTP is lost. However, it should be noted that SRTP does not specify how keys should be distributed to the final clients.

Since key distribution is not standardized, yet required for secure multimedia, there are a number of implementations in use [33]. The simplest method for key distribution is simply defining the key values of the multimedia session inside the SIP message using SDES (Session Description Protocol Security). This method was formally published in RFC 4568 which was published in July 2006 [3]. SDES states that the binary key should be encoded in base-64 so that it is compliant with the Multipurpose Internet Mail Extensions (MIME) as described in the Session Description Protocol (SDP). Since no security for the key exists in the application layer unless S/MIME is used, this is at best a temporary solution for allowing the SRTP to get critical mass within industry. Another major reason for implementation of SDES is the performance of the applications that use SDES for key distribution is greatly improved; making capabilities such as push-to-talk seem more natural to the end user.

An additional option for enabling key distribution is the Multimedia Internet KEYing (MIKEY) technique that was published in RFC 3830 in August of 2004 [5]. Three techniques exist for distributing keys as specified in the protocol; providing a pre-shared key before the SRTP session begins, using a public key distribution system, or implementing the Diffie-Hellman

security approach. The obvious problem with the pre-shared key is that each client must have a series of keys for all of the other clients in the network. This solution does not scale with large networks, and is not used with regularity. Use of a public key distributed using a PKI (Public Key Infrastructure) is another possibility that has seen use in many DoD (Department of Defense) organizations through the efforts of Defense Information Security Agency (DISA). Finally, there exists the Diffie-Hellman technique for key distribution. Since this final option of Diffie-Hellman is a solution that scales without a high overhead cost, it is examined in detail in this section.

To describe how Diffie-Hellman works in relation to two users, it is best to define the two client stations as Alice and Bob as can be seen in Figure 9. A prerequisite to the Diffie-Hellman exchange requires that each of the two users determine their own private key before the exchange begins. After Alice has her key and two other public values,  $g$  and  $p$ , she calculates  $A = g^a \text{ mod } p$  using her private key  $a$ . She next sends her calculated value,  $A$ , to Bob with the values  $g$  and  $p$ . Bob calculates his own public value using the  $g$  and  $p$  value given by Alice in addition to the shared secret by calculating  $K = A^b \text{ mod } p$ . Bob sends his calculated value  $B = g^b \text{ mod } p$  to Alice, so that she in turn can solve  $K = B^a \text{ mod } p$ . In this way, both Alice and Bob have negotiated a shared key for the data encryption.

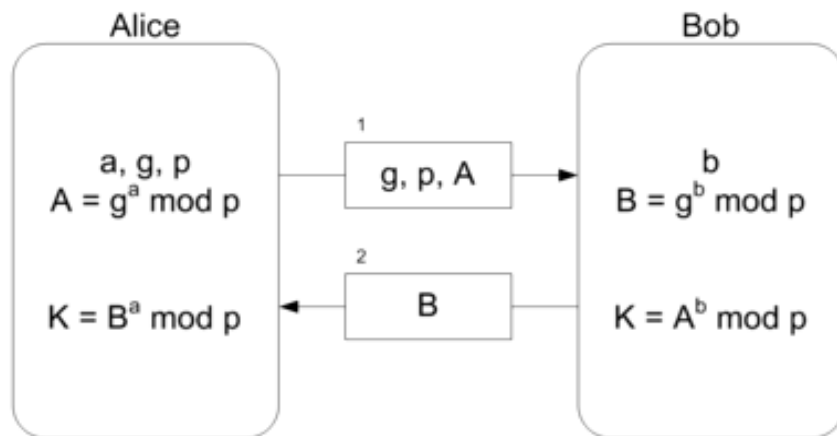
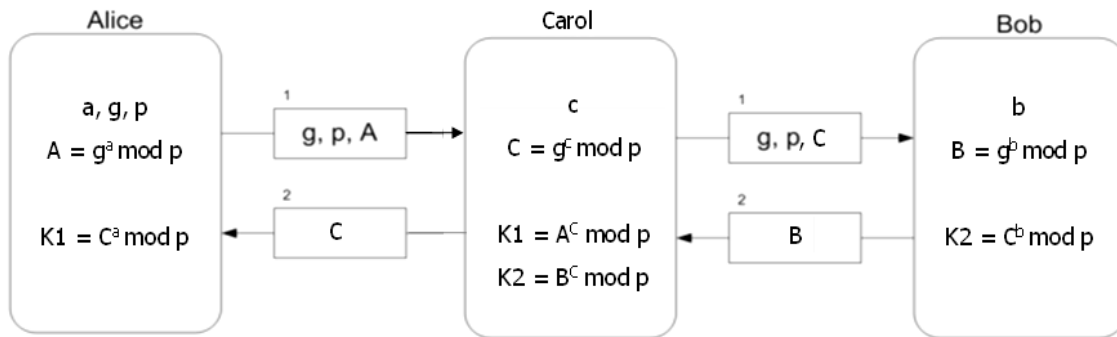


Figure 9. Diffie-Hellman Key Distribution

A major flaw of the Diffie-Hellman Exchange is a vulnerability to a man-in-the-middle attack [28][49]. Specifically, if an intruder exists that is able to intercept both Alice and Bob's messages; the intruder has access to the key for the data encryption. For this example, Carol is the intruder. When Alice inadvertently sends her calculated value  $A$  to Carol, Carol uses the  $g$  and  $p$  values to calculate her value  $C$ . Carol sends  $C$  back to Alice, in this way ensuring that Alice and she are sharing a common key. In the same way, Carol sends the value  $C$  along with  $g$  and  $p$  to Bob who shares a second key with Carol. In this way, Carol uses two keys, one to unencrypt Alice's messages and one to unencrypt Bob's messages.



A solution was proposed in March of 2006 that would eliminate the man-in-the-middle attack through use of a series of shared secrets. This solution is ZRTP (Zimmerman RTP) [50] which was named after founder Phillip Zimmerman in an internet draft. ZRTP is essentially secure unless the intruder was present for all sessions between the two UAs. A series of shared secrets between the two clients is determined and maintained to ensure that no man-in-the-middle attack is occurring. Specifically, each client has its own ZRTP ID value that it shares with other clients and additional side-channel signaling is supported in the protocol. Also, ZRTP relies on keyed-hash message authentication code (HMAC) capability to allow a double Diffie-Hellman message exchange to ensure that both of the clients authenticate one another's message. Only

after successful completion of the second Diffie-Hellman exchange is it possible for a SRTP session to begin.

Details for implementation of the ZRTP protocol are shown in Figure 11. A RTP session must already be in existence for the start of the ZRTP exchange. Also, the ultimate product of the exchange is a SRTP session. The value “ver” stands for the implementation version. “cid” represents the client’s ID which is used for a lookup of the public key; it is associated with a particular software installation. “hash” represents which hash algorithm is used for authentication. “cipher” represents the type of cipher that is used in the SRTP session; generally, this is the AES (Advanced Encryption Standard). “pkt” represents the public key type; this value is represents the length of the key and is usually either 128 or 256 bits. “sas” represents an optional short authentication string that can be used within the exchange to further secure the session. “ZID” represents the ID of the user. These six variables are transmitted in the first two phases of the exchange to allow a secure Diffie-Hellman exchange in the final two phases of the ZRTP protocol exchange.

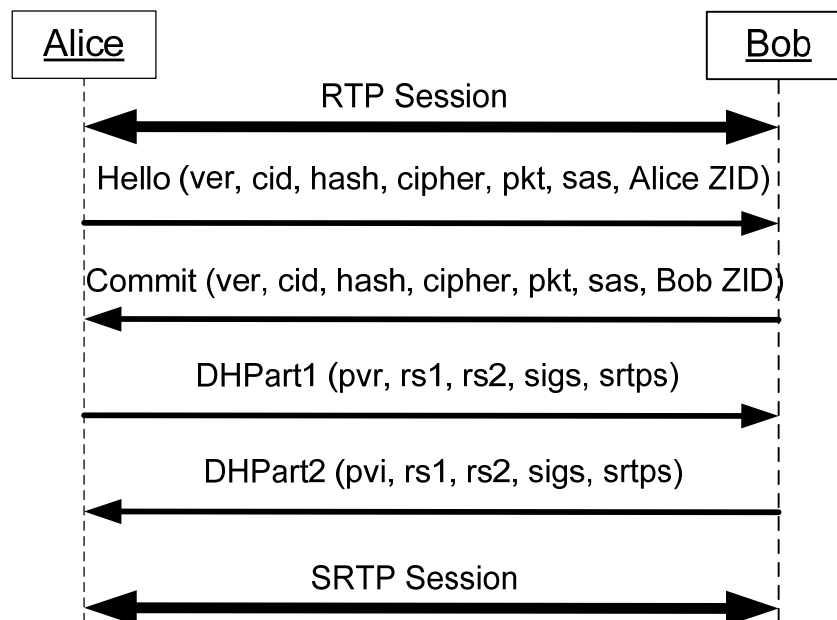


Figure 11. Zimmerman’s Solution to Diffie-Hellman Vulnerability



The information to be used in the SRTP session is passed in the “srtps” value, which represents a hash algorithm on the combined string of the SRTP key and salt. “sigs” represents the signaling secret which is generated by use of the hash algorithm on the combined string of the call-id, as well as the local and remote tags, all of which are used in the SIP layer. “pvr” and “pvi” represent a session-specific secret that should be used to secure the “srtps” value. These values are calculated by the expression:  $g^{svr} \bmod p$ . Where  $g$  and  $p$  are public values and  $svr$  is a randomly chosen number for this session. “rs1” and “rs2” represent out of band secrets, which can be used to further secure the “srtps” value. If the out of band values are not available, random numbers are used instead.

## 2.6 Authentication, Integrity and Replay Protection

The hash algorithm has already been discussed in the implementation of the ZRTP protocol; however it is being defined in this section of the thesis. In addition to the ZRTP protocol, SRTP specifies that the keyed-hash message authentication code (HMAC) be used to enable authentication, integrity and replay protection capability. HMAC functions by combining both the key and the message in one block [7]. The procedure is:

$$HMAC_K(m) = h((K \oplus opad) || h((K \oplus ipad) || m)) \quad (2)$$

where the first step is to conduct an XOR function with  $K$  (the key) and the inside pad “ipad” defined as “0x3636...”. After concatenating the result with the message “ $m$ ”, the term can be hashed using one of several hash algorithms to be discussed below. This hashed result is concatenated with the XOR result of the key,  $K$  and the outside pad “opad,” which is a constant; “0x5c5c...”. Finally, a second hash occurs over the entire result.

Definition of the hash algorithm is not present in the HMAC specification; however, the choice of which hash algorithm to implement has a significant effect on the ultimate security of

the HMAC algorithm. Understanding of how the hash algorithm functions requires knowledge that the hash algorithm produces a fixed sized “digest” for each message input. Despite how small the change to the initial input, the digest varies by seemingly random change. A widely implemented hash function is the Secure Hash Algorithm (SHA) – 1, which uses a 160 bit digest for securing the message and key [12]. However, Vincent Rijmen (one of the inventors of the Advanced Encryption Standard), published a paper in 2005 describing an attack on SHA-1 [38]. For this reason, SHA-1 should not be used as the hash algorithm for HMAC. As of April 2007, digest lengths from 224 to 512 bits as specified in the SHA-2 family of algorithms do not have published attacks. It should also be noted that a salt is needed for generation of the hash value in SHA. It is possible to pass that salt value through a ZRTP procedure or send it by SIP messaging using SDES.

## **2.7 Advanced Encryption Standard**

The SRTP defines AES [1] (Advanced Encryption Standard) to encrypt the data stream. AES was defined by the National Institute of Standards and Technology (NIST) in the US Federal Information Processing Standard (FIPS) in Publication 197 in November of 2001. The AES standard is largely based on a protocol that was submitted by Joan Daemen and Vincent Rijmen under the name Rijndael. AES differs from the Rijndael algorithm by stating that the key size is limited to 128, 192 or 256 bits, whereas the Rijndael standard uses any multiple of 16 bits between 128 and 256 bits. The block size of the algorithm is limited to 128 bits; however, the blocks can be used in one of several modes of operation to extend the length of the message to be encrypted. SRTP defines either the counter method or output feedback stream ciphers for use in encrypting RTP packets, which is discussed following the method for securing a block of information.

256 bit key			
Step 1	Step 2	Step 3	Step 4
4 bytes	12 bytes	4 bytes	12 bytes

Figure 12. Building a 256-bit Key in Four Steps using the AES Key Schedule

The blocks are encrypted in several rounds; the exact number of rounds depends on the key size chosen (e.g. 256 bits for 14 rounds). The Rijndael key schedule requires four steps to determine all values within the needed key size [11]. These four steps and the relationship to each rounds' key is show in Figure 12. On the first round, the first four bytes of the first key are generated by performing an operation called the key schedule core on the first four bytes of the original key. A visual description of the core procedure is available in Figure 13. The key schedule core's first step is to rotate the eight bits (a byte) to the left. Next, the Rijndael S-box operation occurs four times to the shifted 256 bit sequence. The Rijndael S-box function is essentially a table lookup, as can be seen in Table 1.

Table 1. Rijndael S-Box [11]

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

As the final step of the key schedule core, an xor operation is completed using the left most byte with a value called the "rcon". The "rcon" value increases in each key schedule

iteration by a value of  $2^i$  in the Rijndael Finite Field [6], where  $i$  increments on each pass through the key schedule core throughout the key schedule process by a value of one. The calculated value in the Rijndael finite field is based on input  $i$  is the sum  $i^8 + i^4 + i^3 + i + 1$ .

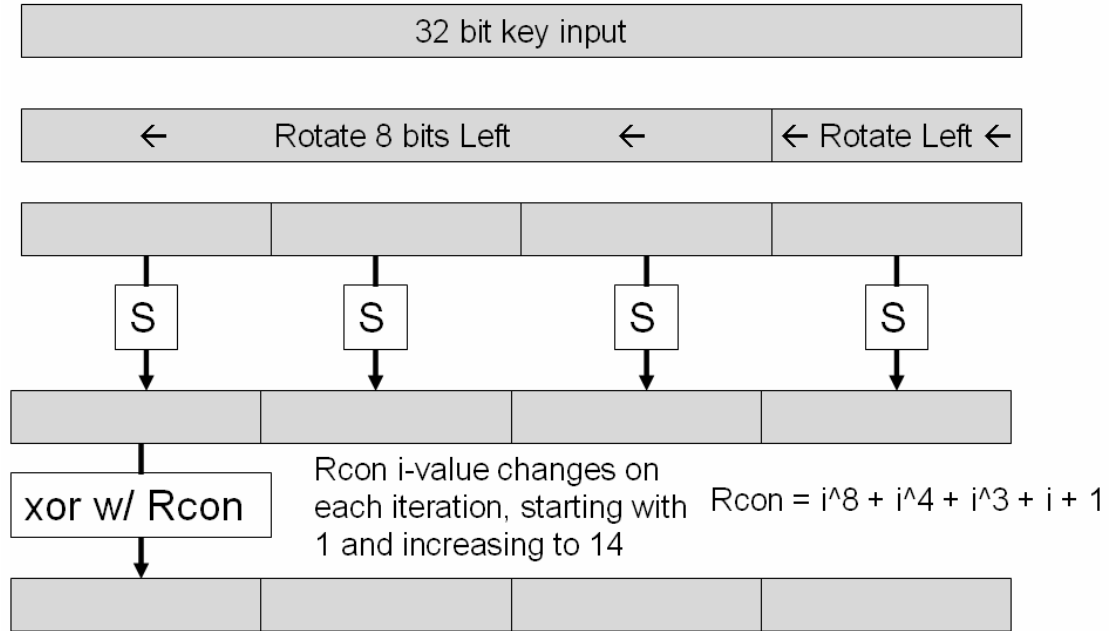


Figure 13. AES Core Procedure for Key Generation

The second step is iterated three times, to generate 12 bytes. The first iteration xors the four bytes that were used in the key schedule core process with the four bytes that are the product of the AES core procedure to generate four bytes. The second iteration xors the product of the core procedure with the product of the first iteration of the second step to generate four bytes. The third iteration xors the product of the first iteration with the product of the second iteration to generate the last four bytes of the second step.

Step three adds another four bytes by conducting the S-Box lookup on the 5<sup>th</sup> four byte word of the original key and xor'ing the result with the four byte product of the third iteration of the second step.

The final 12 bytes are created by iterating through the fourth phase of the process. The first iteration xors the 5<sup>th</sup> four byte word of the original key with the four byte product of third step to generate four bytes. The second iteration xors the product of the third step with the product of the first iteration to generate four bytes. The third iteration xors the product of the first step with the product of the second step to generate the last four bytes of the third step.

After completion of the key scheduling procedure, iterations through the block using each of the keys generated in the key schedule can be completed using the AES procedure [1]. A visual description of this procedure is provided in Figures 14-17 [2]. The first step of a round in AES is to xor each byte of the block with the key as shown in Figure 14.

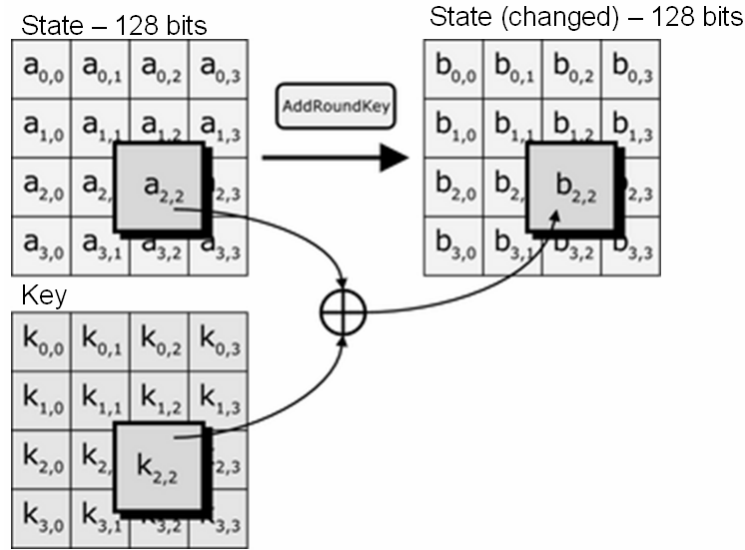


Figure 14. AES Encryption Step One [2]

The second round uses each byte of the output of the first step as the input to an S-Box lookup to generate the output values as shown in Figure 15. The third step shifts the output bytes of step two left as shown in Figure 16. The second row gets shifted by one byte to the left, the third row by two bytes, and the fourth by three bytes. The fourth step performs the following calculations to each of the output block rows based on the equation

$$\begin{aligned}
 r_0 &= 2a_0 + a_3 + a_2 + 3a_1 \\
 r_1 &= 2a_1 + a_0 + a_3 + 3a_2 \\
 r_2 &= 2a_2 + a_1 + a_0 + 3a_3 \\
 r_3 &= 2a_3 + a_2 + a_1 + 3a_0
 \end{aligned} \tag{3}$$

where  $a_i$  is an individual byte within a column as can be seen in Figure 17. The resulting  $r_i$  values are the final output bytes. After completion of all 14 rounds, the output block is encrypted with the 256-bit key.

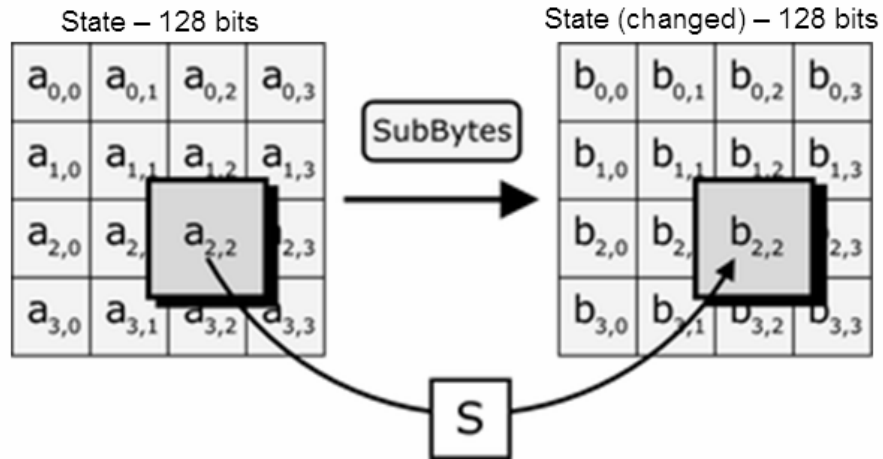


Figure 15. AES Encryption Step Two [2]

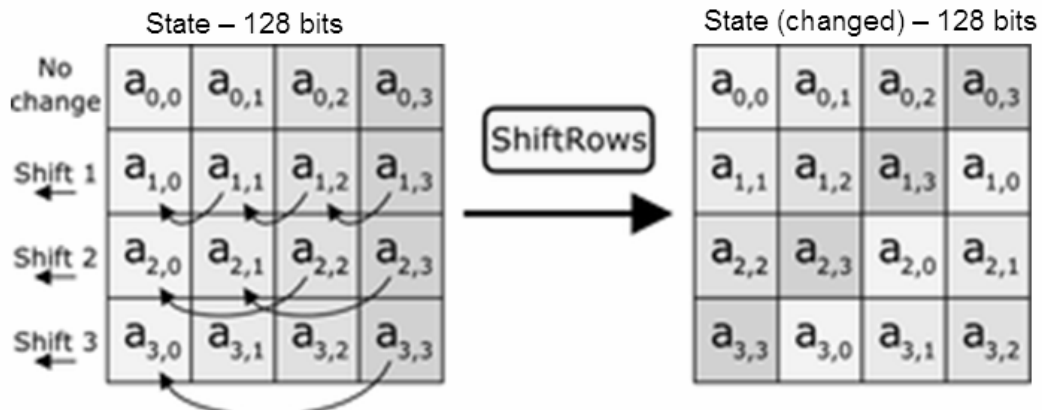


Figure 16. AES Encryption Step Three [2]

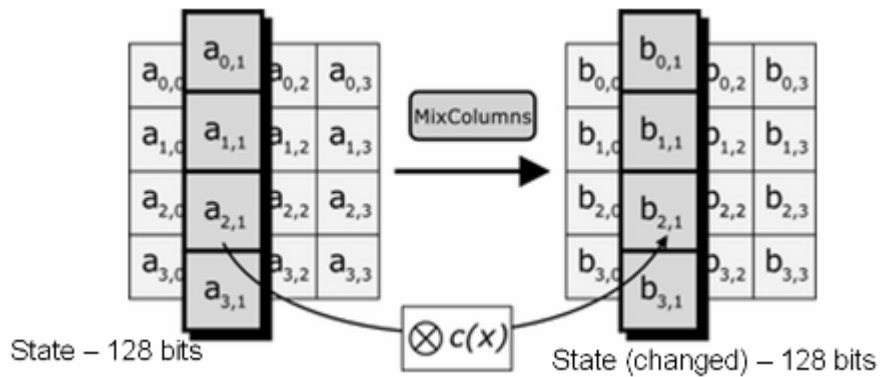


Figure 17. AES Encryption Step Four [2]

## 2.8 Block Cipher Modes of Operation for Secure Real-Time Transport Protocol

A further complexity to consider when implementing SRTP, which must account for streams of data, is how to work with multiple blocks without a change in the key. SRTP suggests one of two methods be used for the block cipher modes of operation - segmented integer counter mode (CTR) or an altered form of the output feedback mode (OFB). In order to use both of these modes it is required that both a key and salt be supplied by either SDES in SIP messaging or through ZRTP session initialization [8].

Segmented integer counter mode (CTR) is the default for AES stream encryption. CTR is differentiated by use of an initialization vector (IV) that increments so that no repeats occur in a short time. The IV is first encrypted by use of the key using the AES algorithm; this is shown in Figure 18 as a “block cipher encryption”. It is then be xor’ed with the plaintext data stream to produce the enciphered data stream that can be packaged within an RTP packet and sent to the destination.

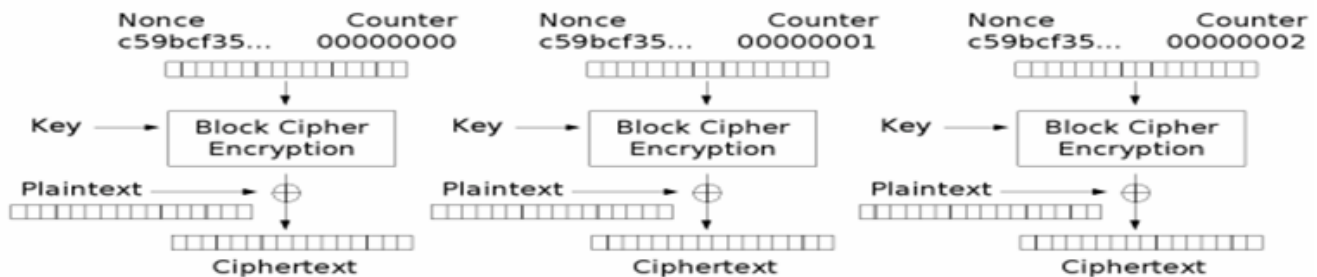


Figure 18. Counter (CTR) Mode Encryption [7]

A second method for encrypting the data stream is through use of an IV that never increments; this process is called the output feedback mode (OFB). As seen in Figure 19, OFB functions by taking the IV, encrypting it using the AES algorithm, and then xor’ing it with the plaintext data stream before packaging it inside an RTP packet. The pattern is repeated by reprocessing the enciphered IV through the AES algorithm a second time and then xor’ing with the plaintext stream to generate the contents of the second RTP packet.

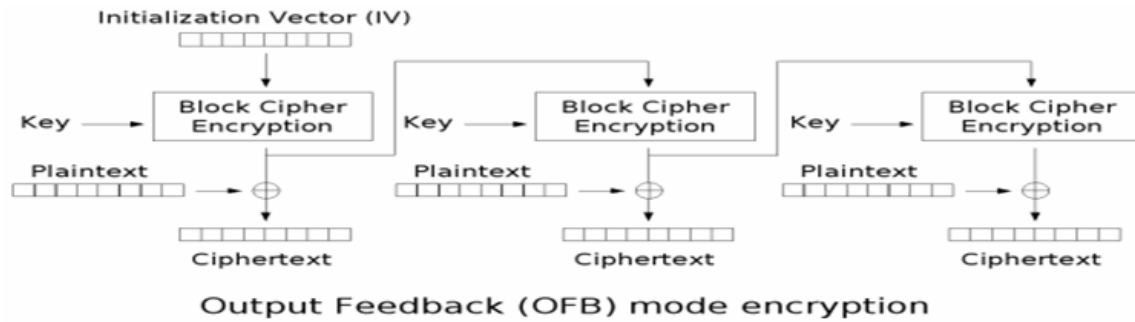


Figure 19. Output Feedback (OFB) Mode Encryption [8]

## 2.9 Methods for obtaining MOS (Mean Opinion Score)

Proper analysis of the multimedia connection requires a well-established method for obtaining perceived quality. One of the methods for obtaining MOS is the use of human subjects to subjectively rate the quality on a scale from one (bad or unusable) to 5 (excellent) [26]. Another way to interpret the results of this study is from the point of view of the impairment that the conditions provide. Table 2 illustrates the relationship between the numerical value and the quality or impairment that was experienced by the users. Another possible method for testing the quality of the signal is use includes the ITU (International Telecommunications Union) P.862 PESQ (Perceptual Evaluation of Speech Quality) [14] or the E-Model [45]. The E-Model uses the rate of packets loss, delay, as well as the inherent codec weaknesses to predict the score that would be obtained if the human subject data was instead collected. The output of an E-Model is generally reported in terms of the R-Value which is an integer from 0 (bad) to 100 (excellent).

Table 2. MOS Quality and Impairment Numerical Relationship [29]

<b>MOS</b>	<b>Quality</b>	<b>Impairment</b>	<b>R-Value</b>
4.5	Excellent	Imperceptible	90
4.0	Very Good	Perceptible, but not annoying	80
3.5	Good	Very slightly annoying	70
3.0	Fair	Slightly annoying	60
2.5	Poor	Annoying	50
1.0	Bad	Very annoying	0



For comparison with the results that are analyzed in Chapter IV, the MOS scores of several codecs are shown in Table 3. The G.711 standard is used in this thesis since it has the highest MOS score.

Table 3. MOS Scores of Various Codecs [29]

Codec (data rate)	Mean Opinion Score (MOS)
G.711 (64 kbit/s)	4.1
G.729 (8 kbit/s)	3.9
G.723.1 (6.3 kbit/s)	3.9
G.729a (8 kbit/s)	3.7
G.723.1 (5.3 kbit/s)	3.7

## 2.10 Related Research

Several analytical and computer-based models of wireless multimedia systems exist that can be compared to the results obtained in this thesis. Additionally, existing data for wireless multimedia systems exist as well. This section describes the various models and data sets.

Multiple models to describe the use of the wireless medium for purely voice traffic were developed before this study [10,13,16]. The model in [17] is given by

$$N = \frac{T_{vp}}{T_{transmit}} \quad (4)$$

where  $T_{vp}$  is the size in ms of the voice payload, and  $T_{transmit}$  is the time involved in transmitting a packet. The  $T_{transmit}$  is further defined as

$$N = 2 * (T_{voice} + SIFS + T_{ack} + DIFS) + (T_{slot} * CW_{min} / 2) \quad (5)$$

where  $T_{voice}$  is the time taken to transmit a packet, SIFS is the short inter frame space,  $T_{ack}$  is the time for the receiver to acknowledge receipt of the packet, DIFS is the DCF inter frame space,  $T_{slot}$  is the discrete wait interval length and  $CW_{min}$  is the maximum number of intervals [27]. Equations 4 and 5 are reformed in [44] using:

$$N = \left\lfloor \frac{1}{\sum T_i} \right\rfloor \quad (6)$$

where capacity (N) is a function of the sum of percents where each activity within a node is denoted by a percent of total capacity ( $T_i$ ). Activity  $T_i$  can be expressed as one of two types – back-off after the transmission of a packet and the actual transmission of a packet. A possible activity that a node can be contributing to total network activity is engaging in back-off activity related to idle time in the network following the transmission of a packet. The percent of network capacity expended by a single back-off procedure following the transmission of a type of packet is

$$T_{BACKOFF} = R_{MEDIA} (T_{SLOT} \times CW_{MIN} / 2) \quad (7)$$

where  $R_{MEDIA}$  is the rate of real-time packets and the values  $T_{SLOT}$  and  $CW_{MIN}$ .

Recommendations of the values to be used within equation 7 are in Table 4.

Table 4. Recommended Values for  $T_{BACKOFF}$  Calculation

Format	Value	Description
$R_{MEDIA}$	50 packets/sec	Rate of Packets
$T_{SLOT}$	20 $\mu$ s	Discrete Wait Interval Length
$CW_{MIN}$	31	Max Number of Intervals

A general expression for the percent of network capacity that is used by one type of media transmission is

$$T_{TX} = 2R_{MEDIA} (T_{MEDIA} + SIFS + T_{ACK} + DIFS) \quad (8)$$

where  $T_{MEDIA}$  is the media transmission time of the RTP packet, SIFS is the short inter frame space,  $T_{ACK}$  is the time for the receiver to acknowledge receipt of the packet and DIFS is the DCF

inter frame space [27]. Table 5 provides recommendations of audio and video values that can be used to calculate  $T_{TX}$ .

Table 5. Recommended Values for  $T_{TX}$  Calculation

Variable	Audio	Video	Description
$R_{MEDIA}$	50 pkts/s	50 pkts/s	Rate of packets
$T_{MEDIA}$	72.8 ms	158 ms	Media Tx Time
SIFS	10 ms	10 ms	Inter frame Space
$T_{ACK}$	41.2 ms	41.2 ms	ACK Tx Time
DIFS	50 ms	50 ms	DCF Inter frame

The results of calculating the capacity of the network using Equations 6,7 and 8 with the values in Tables 4 and 5 is shown in Table 6. The 176x144 Video Payload [47][48] condition is predicted to support 19 nodes, or 9 conversations. Also, computer simulation data in [17] suggested that capacities of four conversations are possible in an 802.11b WLAN.

Table 6. Analytical Model of Audio-Video Capacity

Screen Size (Pixels)	Number of Nodes on 802.11g WLAN
128x96 Video Payload	20
176x144 Video Payload	19
320x240 Video Payload	14
352x288 Video Payload	11
352x480 Video Payload	3

Analysis and simulation completed in [17] and [44] both made the assumption that packet loss rates are less than 3 percent. However, empirical analysis in [40] has been demonstrated that the packet loss percentages in 802.11g WLANs often have loss rates greater than 50% over significant (> 5 seconds) portions of an extended connection.

Data in [40] was also related back to audio MOS scores through use of neural network designed to imitate human reaction to an audio signal. The data suggests that there are

continuous periods of fifteen seconds or more where the audio MOS is less than 2 out of 5 (poor).

This suggests that not a single VoIP conversation can occur on an 802.11g WLAN.

With such great differences in the wireless network literature regarding the capacity of 802.11g to support voice and video traffic, it is reasonable to collect and analyze actual human subject data on the voice and video capacity available through a secure wireless system.

## **2.11 Conclusion**

This chapter describes the current state of VoIP technology with special focus on current methods to secure the real-time traffic. Section 2.1 discusses two families of protocols capable of providing voice and video services over IP networks - H.323 [25] and SIP. Section 2.2 describes how RTP is used to transmit the data containing the real-time voice and video data. Section 2.3 describes how audio information is compressed so that it may be transmitted using RTP. Section 2.4 describes QoS improvements to the data link layer currently implemented in the IEEE 802.11e standard. Section 2.5 describes how to transmit keys for the purpose of encrypting the real-time traffic. Section 2.6 describes how authentication, integrity and replay protection schemes are enabled by the keyed-hash message authentication code. Section 2.7 describes how AES is used to encrypt the real-time content. Section 2.8 describes how the block cipher modes of operation use AES to secure a stream of real-time traffic. Section 2.9 describes how MOS are obtained and how they relate to the quality of the audio and video traffic. Section 2.10 describes related research initiatives for determining the capacity of wireless audio-video systems.

### **III. Methodology**

This chapter describes how the experiment is designed as well as the method to set up the laboratory software. The two primary questions that are addressed in this chapter are: 1) Is there a statistically significant effect ( $> 95\%$  chance that differences in the dependant variables result from manipulation of the factor) on the quality of the audio and video data by routing the traffic through an AP compared with two arbitrary nodes on a WLAN transmitting and receiving audio and video traffic?; and 2) Is there a statistically significant impact on the quality of the audio received by securing the audio traffic? Section 3.1 describes the experimental design based on research goals. Additionally, a method that was developed as part of this thesis effort to secure multimedia IP traffic when using Java applications is discussed in detail. Section 3.2 describes the high level architecture of software that is used for collection of the human subject data. Section 3.3 describes how the wireless network is secured against network layer compromises of security. Section 3.4 summarizes the methodology development that is explained within this chapter.

#### **3.1 Experimental Design**

Several effects of variables related to secure wireless multimedia are addressed in this thesis. One of the variables manipulated during the course of the thesis experiment is whether the voice is secured. The thesis experiment also is designed to determine how many voice and video connections are possible, and at what network conditions the signal quality degrades enough to become annoying to users of the system. Also, the impact of sending the signal through an AP

has as opposed to having two terminals on the same WLAN is part of the design of the thesis experiment.

These research goals are answered by exposing subjects to several network conditions and then measuring the MOS (Mean Opinion Score) of the entire subject pool. Since human subjects are used to understand the effect of different wireless multimedia configurations, an exemption of the subject protocol is obtained. Since all of the subject data is collected in WPAFB Area B Bldg 190, which houses the Logistics Readiness Branch, (AFRL/HEAL) of the Air Force Research Laboratory (AFRL), the exemption was sought through the AFRL process. Based on the paperwork required for the exemption process, a human subject consent letter (Appendix A) and an information form (Appendix B) are created. The letters are signed to confirm that the subject has knowledge of their rights under the Privacy Act Laws. Also, an information form is available for each subject to retain in their possession which informs the subject to the Privacy Act Laws regarding their participation in the study.

A possible confounding variable in the study is a possible existence of subjects that have moderate to severe hearing loss. For this reason, all subjects have a short hearing screening test to ensure that all subjects have minimal (20 dB or less) hearing loss. The Home Audiometer Hearing Test [18] is used to conduct the hearing screening test. To begin the screening test the subject is shown how to manipulate the joystick to indicate they hear the sounds created by the Home Audiometer Hearing Test.

A list of the different network conditions that the thesis software produces is detailed in Table 7. As an example of how to read the descriptions in Table 7, the network condition two is defined as two-way stream flowing from the laptop through an AP to the subject. Figure 20 illustrates the network conditions that are listed in Table 7. Network conditions three and four within the audio and video media type are excluded because the multimedia signal was unquestionably unusable.

Table 7. Network Conditions

Media Type	Network Condition	Description
Just Audio	1	laptop => AP => subject
	2	2 * (laptop => AP => subject)
	3	3 * (laptop => AP => subject)
	4	4 * (laptop => AP => subject)
	5	laptop => subject
	6	2 * (laptop => subject)
Audio and Video	1	laptop => AP => subject
	2	2 * (laptop => AP => subject)
	5	laptop => subject
	6	2 * (laptop => subject)

Three of the four variables in the experiment are within-subject variables, meaning that the factors are experienced at different levels by each of the subjects. As an example, the use of video factor is experienced in the on and off levels by each subject. For this reason, the use of video is a within-subject variable. Additionally, the number of active conversation and use of the AP are within-subject variables. The remaining factor is the use of security because one subject in the study either experienced the secure audio stream or did not. For this reason, the two levels of the factor were only different between subjects; therefore, it is referred to as a between-subject variable. The reason that the use of security factor is a between-subject variable is that the experiment was initially envisioned to either have a secure wireless network (with WPA encryption) or an open wireless network to compare the effects, which could not have been easily reconfigured during a subject run. However, in the final iteration of the software it is possible to turn the security feature on and off instantaneously.

All human subjects are first exposed to the audio only conditions, which consist of six network conditions as described in Table 7 and illustrated in Figure 20. After exposure to the audio signal, the software prompts the user rate perceived quality of the audio sample. Next, the subjects are exposed to the video conditions in the same order as in the audio conditions as shown

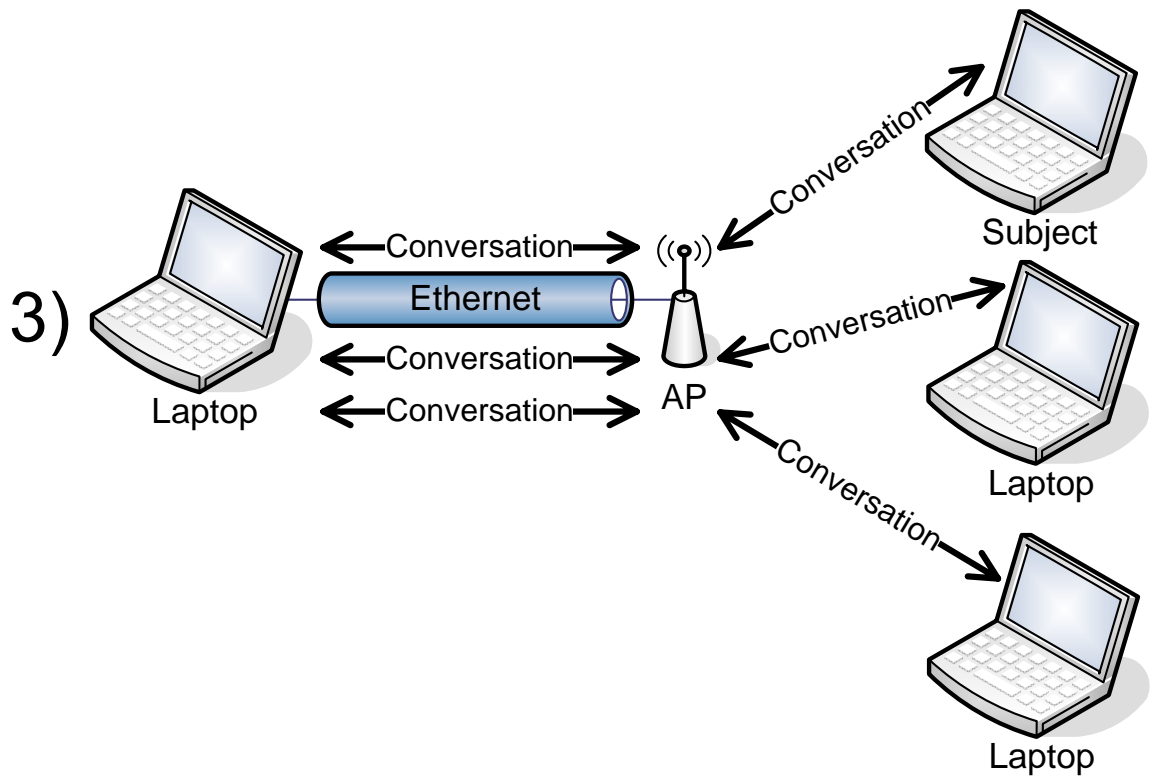
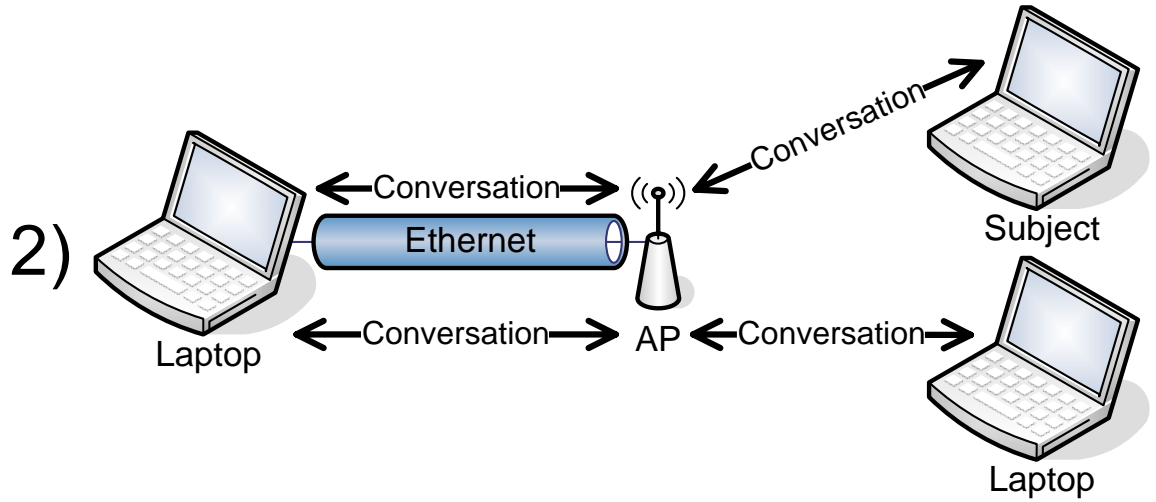
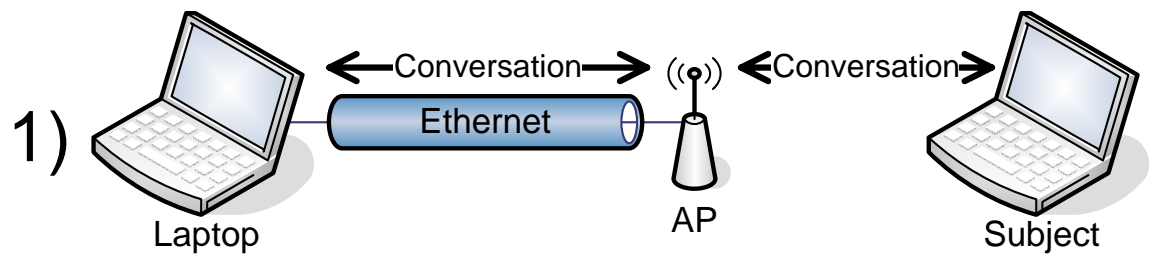


Figure 20. Network Conditions



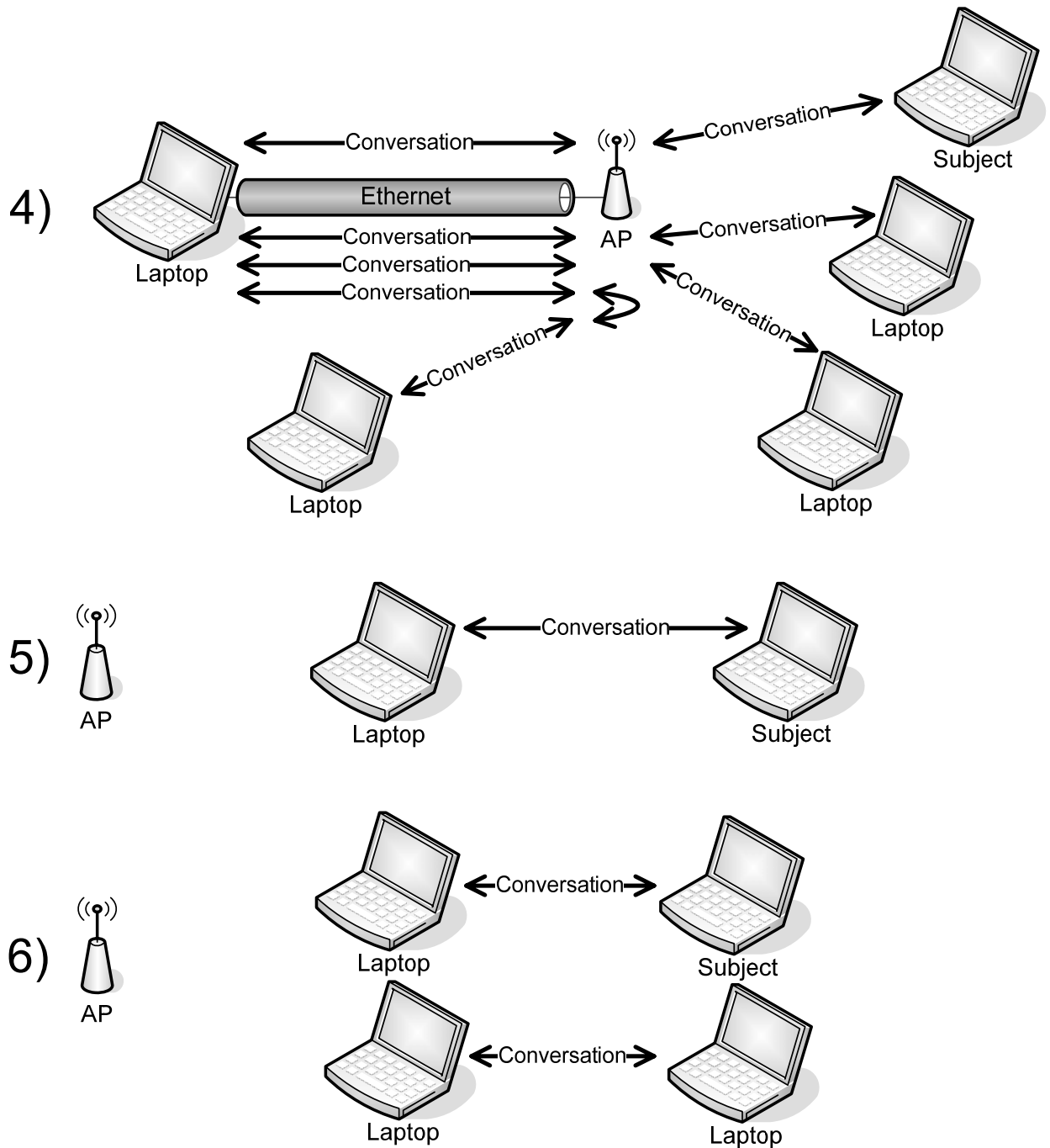


Figure 20. Network Conditions (continued)

in Table 7. The subjects are then prompted by the software to evaluate both the audio and video in terms of perceived quality. Once the human subject has provided inputs to both values, the subject presses the “OK” button to continue to the next network condition.

To determine the correct number of subjects to use in the study, this equation was used

$$n \approx z_a^2 \frac{\sigma^2}{\delta^2} \quad (9)$$

where n is the appropriate number of subjects  $z_a$  represents the appropriate deviation for mean for the 95% confidence interval which is evaluated to be 1.96. The  $\sigma^2$  is the expected standard deviation of the sample data set (0.9) and  $\delta$  is half the size of the final confidence interval (.1) [32]. The needed number of subjects is evaluated at thirty six.

All thirty-six subjects are randomly assigned to one of these twelve different conditions to ensure that the experimenter does not provide a confounding variable related to the order and treatment of the subjects. Six of these conditions use a secure voice connection, and six of these conditions do not use the secure feature. In order to ensure that there exist no ordering effects among the network conditions, the order of the network conditions is counterbalanced over the six different network configurations. Figure 21 illustrates an approximate view of how the user interface for the MOS evaluation appears.

Estimated Voice Quality	Estimated Video Quality
<input type="radio"/> 5. Excellent	<input type="radio"/> 5. Excellent
<input checked="" type="radio"/> 4. Good	<input checked="" type="radio"/> 4. Good
<input type="radio"/> 3. Fair	<input type="radio"/> 3. Fair
<input type="radio"/> 2. Poor	<input type="radio"/> 2. Poor
<input type="radio"/> 1. Unusable	<input type="radio"/> 1. Unusable

Done

Progress Bar

Figure 21. Software User Interface for Audio/Video MOS

In numerical order, the conditions include:

1) A human subject sending and receiving a real-time signal with a laptop connected by Ethernet to the AP.

2) A human subject sending and receiving a real-time signal with a laptop connected by Ethernet to the AP, while at the same time the same laptop connected by Ethernet to the AP is sending and receiving one additional signal with a third laptop on the WLAN.

3) A human subject sending and receiving a real-time signal with a laptop connected by Ethernet to the AP, while at the same time the laptop connected by Ethernet to the AP is sending and receiving two additional signals to a third and fourth laptop on the WLAN.

4) A human subject sending and receiving a real-time signal with a laptop connected by Ethernet to the AP, while at the same time the laptop connected by Ethernet to the AP is sending and receiving three additional signals to a third, fourth and fifth laptops on the WLAN.

5) A human subject sending and receiving a real-time signal with a laptop on the same WLAN.

6) A human subject sending and receiving a real-time signal with a laptop on the same WLAN, while at the same time a two additional laptops on the network are sending and receiving a real-time signal to one another.

For the audio conditions, all six network configurations are used. However, for the video conditions, the third and fourth network conditions are not evaluated. These two network conditions were eliminated from study since it was discovered during the initial software development that there exists unusually poor quality of the video due to technology limitations in the 802.11g wireless networks. The unevaluated perceived video quality of these two network conditions is often poor enough that no video can be successfully received, or just a few frames arrive in the time space of fifteen seconds.

One of the problems when using human subject data is that there may be ordering effects in the data set [5]. Ordering effects occur when the subject rates the perceived quality of a audio or video stream based partially on their exposure to an earlier audio or video stream. To mitigate the ordering effect, it will be necessary to use a counterbalancing design to ensure that the order in which the streams were presented to the subject did not affect the perceived quality rating. To produce a counterbalanced design, a Latin square must be used to ensure that each of the six network condition is administered equally in the six possible conditions. A Latin square is produced by filling a  $n \times n$  table with  $n$  different symbols in such a way that each symbols occurs exactly once in each row and exactly once in each column. When these six network conditions are counterbalanced into six different conditions using a six by six Latin square, the ordering of the network conditions is:

- 1) 1,2,6,3,5,4
- 2) 2,3,1,4,6,5
- 3) 3,4,2,5,1,6
- 4) 4,5,3,6,2,1
- 5) 5,6,4,1,3,2
- 6) 6,1,5,2,4,3.

Table 8 illustrates the order in which each of the network conditions is the network conditions are applied consistently with a counterbalanced design. Since the use of security

Table 8. Subject Treatments

1	126354	10	453621	19	<b>126354</b>	28	<b>453621</b>
2	231465	11	<b>564132</b>	20	<b>231465</b>	29	564132
3	<b>342516</b>	12	<b>615243</b>	21	342516	30	615243
4	<b>453621</b>	13	126354	22	453621	31	<b>126354</b>
5	564132	14	231465	23	<b>564132</b>	32	<b>231465</b>
6	615243	15	<b>342516</b>	24	<b>615243</b>	33	342516
7	<b>126354</b>	16	<b>453621</b>	25	126354	34	453621
8	<b>231465</b>	17	564132	26	231465	35	<b>564132</b>
9	342516	18	615243	27	<b>342516</b>	36	<b>615243</b>

features is a between subject variable only half of the subject will have the treatment. Subjects who have a treatment marked in bold within the table indicate that the subject should be exposed to secure audio. Items in italics indicate that for the speech intelligibility section of the experiment a combined audio-video signal is used for the first twenty rhyming words in the final step. The second set of twenty rhyming words use an audio-only stream. However, if the item is not italicized, this indicates that the first twenty rhyming words in the final step are transmitted with an audio-only signal. The second set of twenty rhyming words is then transmitted with a combined audio-video signal.

During the final step of the experimental procedure, the subjects listen to twenty different words over the connection in either the audio-only or combined audio-video condition based on whether the item is italicized. For the second set of twenty rhyming words, the subject will listen to the opposite condition as the first twenty words. For each rhyming word that is spoken over the connection, the subject will match the word that was said to one of six words that are shown on a user interface as illustrated in Figure 22. Halfway through the list of words that are to be sent over the connection, a two-way video connection is to be created or destroyed, based on the assignments in Table 8.

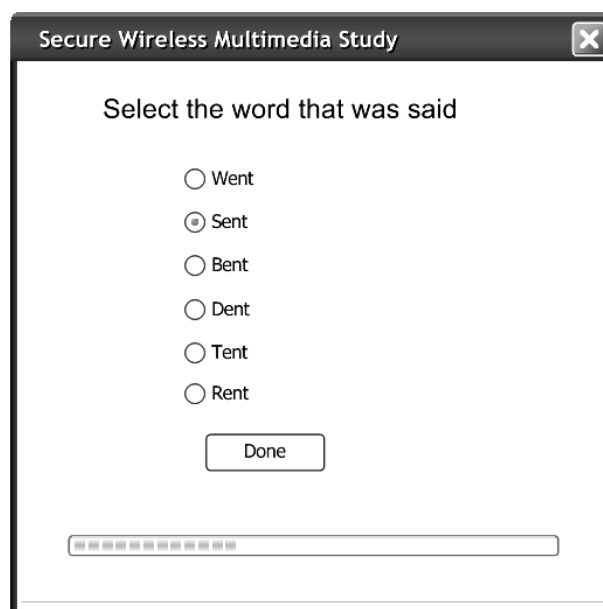


Figure 22. Software User Interface for Audio Intelligibility

### **3.2 Architecture of Experimental Software**

To conduct the experiment, a research platform is developed that is capable of providing both a secure and unsecure real-time audio and video signal from a remote terminal. Since the research platform is unique to this thesis, it is appropriate to describe in detail the development of the underlying software.

As is discussed in the background chapter of this thesis, a two-layered application is generally considered the best design for VoIP-type capability. One of the layers uses the Real-Time Protocol (RTP) to send and receive the data related to video or audio. Another layer is referred to as the top layer and controls the RTP-stream in the lower layer. The top layer performs such functions as the stopping and starting of transmission, and the logging on and off users. The type of data sent within the RTP packets is also communicated between clients at this level. The entire capability is referred to as L3AV, or the Logistics Living Lab Audio-Video capability. Appendix E contains details of the L3AV architecture.

### **3.3 Wireless Network Setup**

Aside from encrypting the application layer data using SRTP, there exists multiple ways to secure the entire wireless local area network [37]. For the purposes of this study, well-established procedures are followed by securing the network at the link layer. The first step in securing a wireless network is to choose a strong SSID (Service Set Identifier). For the purposes of this study, a combination of letters, numbers and symbols with a length of 8 characters is used. Also, the SSID broadcast feature is turned off to make it more difficult to detect the presence of the network. A strong password with upper and lower case numbers, characters and numbers, and at least 12 characters in length is chosen for the administrator password to the wireless router. Additionally, a MAC (Medium Access Control) filter is used. Proper settings for the MAC

address is to allow by exception, therefore, the MAC addresses of all the network cards in use during the thesis experiment are allowed by exception to access the WLAN.

Finally, all network cards are checked for the highest level of encryption technology. The highest level of common encryption technology is the WPA (Wi-Fi Protected Access) Personal encryption technology implemented in the link layer. The advantages of WPA over WEP include the implementation of TKIP (Temporal Key Integrity Protocol). TKIP allows the protection of the key by renegotiation of the key every 10,000 packets. A strong password should be also be used for key generation within the WPA Personal encryption system.

### **3.4 Conclusion**

This chapter describes the design of the experiment and the method to set up laboratory software to address the research goals. The two primary goals that are addressed in this chapter are: 1) Is there a significant effect on the quality of the audio and video data by routing the traffic through an AP compared with two arbitrary nodes on a WLAN transmitting and receiving audio and video traffic?; and 2) Is there a significant impact by securing the audio traffic on the perceived quality of the audio received? Section 3.1 describes the experimental design based on the questions that are of interest to the capacity of secure wireless multimedia. This section discusses a method to secure multimedia IP traffic when using Java applications. Section 3.2 describes the high level architecture of software that is used for collection of the human subject data. Section 3.3 describes how the wireless network is secured against network layer compromises of security.

## **IV. Results and Analysis**

This chapter discusses the results and analysis of the subject data as prescribed in Chapter 3. Section 4.1 discusses how the analysis proceeds throughout the entire chapter. Section 4.2 discusses the results and analysis of the audio MOS data. Section 4.3 discusses the results of the video MOS data. Section 4.4 discusses the results and analyzes the intelligibility data. Section 4.5 summarizes the results of this chapter.

### **4.1 Analysis of Significance**

The raw data collected for this thesis effort can be found in Appendix C. For the significance tests of the factors of interest, SPSS 14.0 for Windows Release 14.0.0 (5 Sept 2005), specifically, the repeated measures feature of the general linear model for MANOVA-type analysis was extensively used. The reason for the widespread use of this technique is that the factor significance can be quickly found by inspecting the p value for all effects (single- and multiple-way) [9]. For details of this analysis technique, please refer to the SPSS documentation [46]. The analysis of the data starts with a general 2x2x2x2 factor analysis and proceeds to more precise data sets to answer the thesis questions in greater detail. All SPSS analysis to determine factor significance can be found in Appendix D.

The research goals will be accomplished through answering five questions:

1) Does the data show significance in terms of number of conversations? Demonstration of significant differences in audio MOS resulting from differences in the number of conversations is important to provide credibility to the collected data set.

2) Is there a significant impact on the audio stream by presence of a video transmission?

This question is similar to Question 1 in that if it is determined that the data shows a significant



decrease in perceived audio quality when the video stream is added, then the data will have greater credibility.

3) Does having one of the terminals connected to an AP by Ethernet have a significant effect as opposed to having only non-AP terminals on a WLAN? If questions 1 and 2 are both answered in the positive, suggesting internal validity, a meaningful result may be determined.

4) Does securing the audio stream significantly affect the MOS of either the audio or video streams? This question is central to this thesis as the effect of securing RTP packets on a wireless network is not known.

5) Does the video MOS data suggest similar effects by the variation in questions 1-4 when compared with the audio MOS? Similar to the audio MOS analysis, if the answer to questions 1 and 2 answered from the perspective of video MOS is positive, then the results of questions 3 and 4 have additional credibility.

For the purpose of expressing the significance of the factors varied in the experiment, the following notation is used:

$$(F_{(a,b)} = x, p = y) \quad (10)$$

where F denotes that the significance is expressed in terms of the F-ratio, a is the degrees of freedom of the factor, b is the degrees of freedom of the error term, x is the calculated value of the F-ratio, p indicates that y is the decimal value indicating significance. If the p-value is below .05, the effect is significant [9].

## 4.2 Audio MOS Analysis

### 4.2.1 Analysis of All Audio MOS Data

When considering the effect on the audio MOS with respect to the within-subjects factors of the number of conversations, use of the AP and use of video in addition to the between-

subjects factor of use of the secured audio RTP, it is determined that the number of conversations ( $F_{(1,34)} = 14.34, p = .001$ ), use of the AP ( $F_{(1,34)} = 13.47, p = .001$ ) and use of video ( $F_{(1,34)} = 99.45, p < .0005$ ) are all significant. However, the between-subject effect of securing the audio stream is found to be insignificant ( $F_{(1,34)} = 3.35, p = .076$ ).

By looking at the two-way effects it is found that the number of conversations by the addition of video traffic has a significant effect on the audio MOS ( $F_{(1,34)} = 5.28, p = .006$ ). This effect is graphed in Figure 35. As an example of how to read Figure 35, consider the slope of the audio-only line labeled as “isVideoTraffic – false.” Note that this slope is steeper compared to the combined audio-video stream as the two-way statistic suggested. Thus, this figure illustrates that increasing the number of conversations has a greater negative impact on perceived quality of the audio-only stream (Audio MOS) when compared with the combined audio-video stream.

Top level analysis of the audio MOS data suggests positive results to questions 1 and 2,

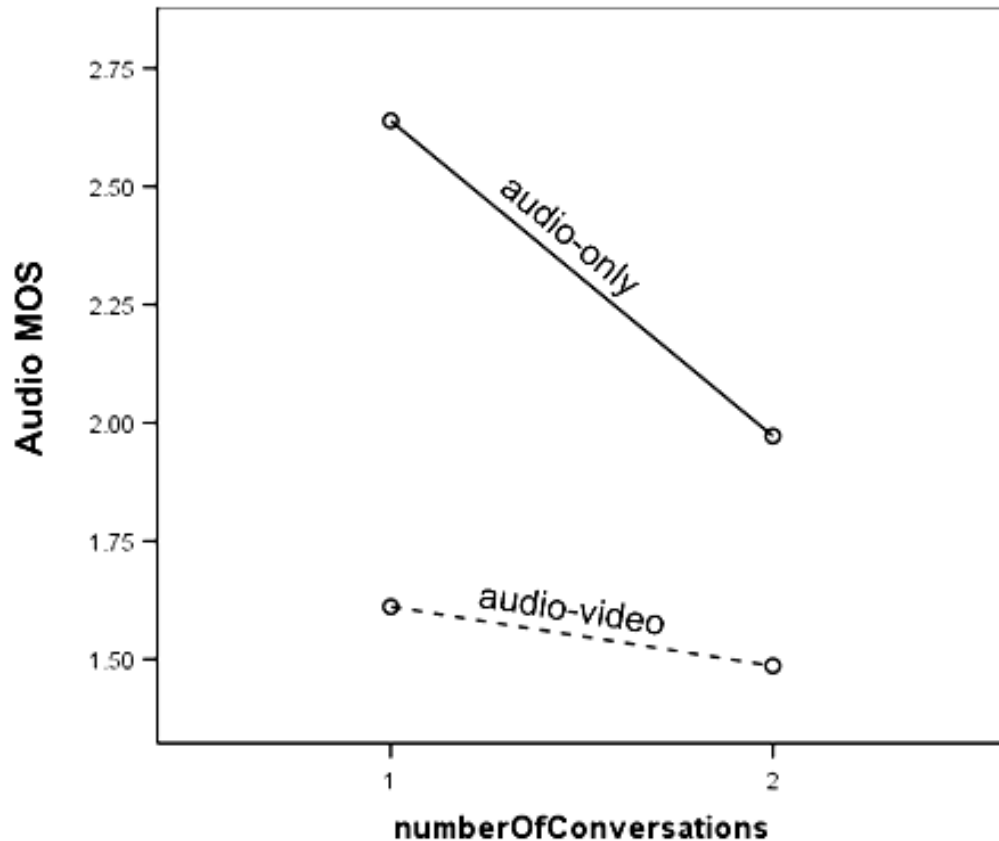


Figure 35. 2-Way Effect Between the Number of Conversation and Use of Video

which are that the number of conversations and use of video in conjunction with the audio stream has a significant effect on the audio MOS. There exists a significant within-subject effect of the two variable number of conversations and the addition of video there also exists a two-way effect between these variables. Question 3 addresses the impact of use of the AP, which was found to be significant, since the audio MOS increased when the multimedia traffic was routed through the AP ( $F_{(1,34)} = 10.49, p = .003$ ).

There also exists a three-way effect between the number of conversations, the use of video, and securing the audio stream ( $F_{(1,34)} = 8.51, p = .013$ ). To view this three-way effect, the two-way effects of number of conversations by use of video traffic is viewed separately when not securing the audio stream in Figure 36 and when securing the audio stream in Figure 37. An immediate explanation of the existence of this three-way effect is not readily available so to understand the circumstance of this three way effect, the with- and without-AP data is analyzed. All other two-way and three-way effects are found to be insignificant as noted in the Appendix D.

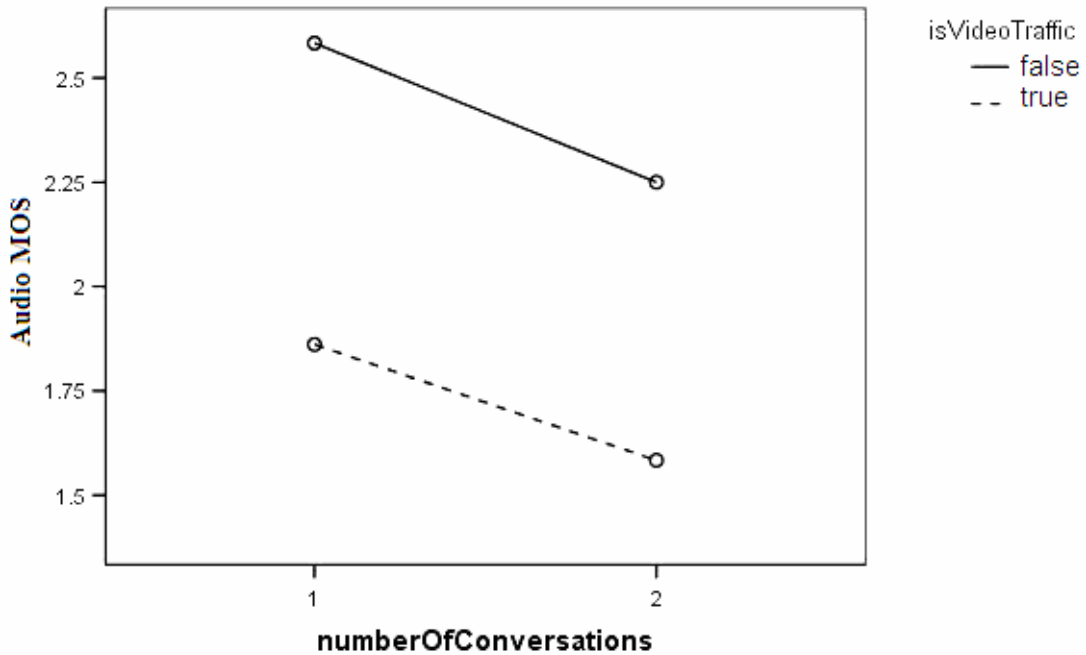


Figure 36. 2-way Effect Between the Number of Conversations and Use of Video (Unsecured)

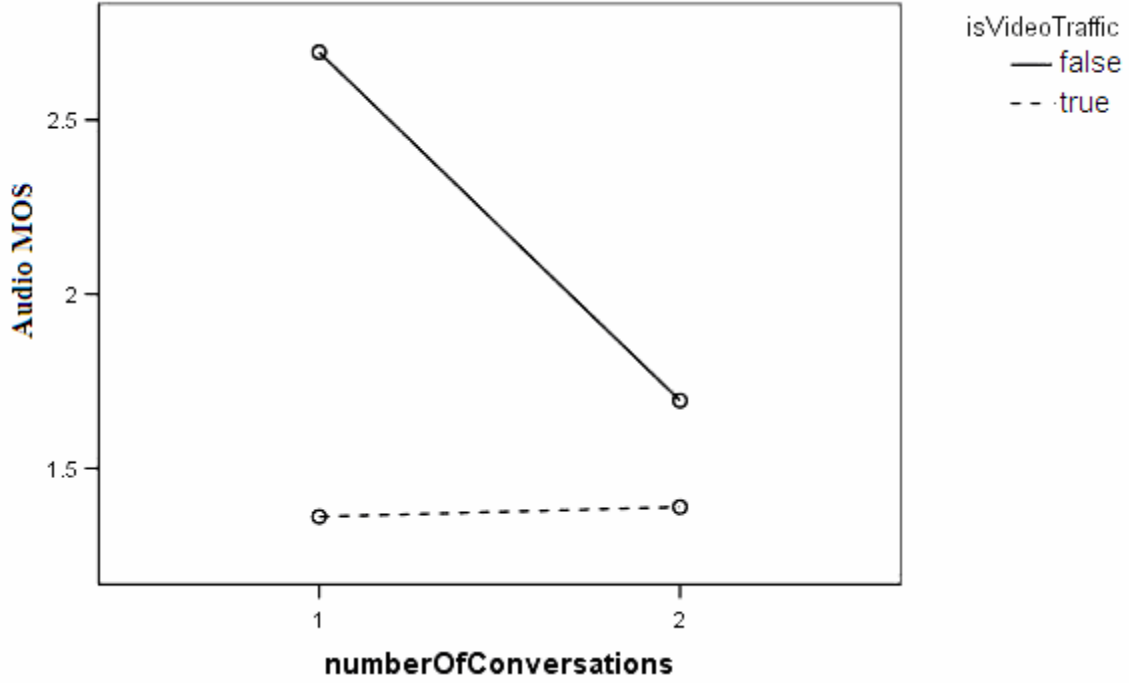


Figure 37. 2-way Effect Between the Number of Conversations and Use of Video (Secured)

#### 4.2.2 Analysis of the With- and Without-AP Audio MOS Data

The use of AP data set indicates similar results to the combined data set in terms of the three way effect of number of conversations by the use of video by securing the audio stream as this effect is still significant ( $F_{(1,34)} = 8.29, p = .007$ ). Also, the AP data set is consistent in terms of the significance in terms of the factors of the number of conversations

( $F_{(1,34)} = 10.00, p = .003$ ), presence of video traffic ( $F_{(1,34)} = 41.65, p < .0005$ ) and the two-way effect of number of conversations by the addition of video traffic ( $F_{(1,34)} = 8.29, p = .007$ ).

Consistent with previous analysis, the securing the audio stream variable does not have a significant between-subjects effect ( $F_{(1,34)} = 2.58, p = .117$ ).

The non-AP data set indicates different results than the combined data and the with-AP data set in terms of the three-way effect as this effect is found to be insignificant

$(F_{(1,34)} = 2.67, p = .111)$ . However, consistent with prior analysis the two-way effect of number of conversations by the use of video is still significant  $(F_{(1,34)} = 34.50, p = .01)$  as is the number of conversations  $(F_{(1,34)} = 10.93, p = .002)$ , use of video  $(F_{(1,34)} = 34.50, p < .0005)$ , and securing the audio stream is not significant  $(F_{(1,34)} = 2.53, p = .121)$ .

The results observed in the non-AP data set contradicts the significance evidence found in the combined and AP-only data only in terms of the significance of the three-way effect of number of conversations by use of video by the variable of securing the audio stream. In later analysis of one and two conversation data and the with- and without- video traffic data sets, there is a specific set of circumstances that may have resulted in this three-way effect, possibly indicating that the three-way effect observed in the overall data and AP-data may have been the result of an experimental design error.

#### ***4.2.3 Analysis of the One Conversation Audio MOS Data***

The relationship of one conversation audio MOS with the within-subjects factor of use of the AP and use of video with the between-subjects factor of use of the secured audio is next studied. A new effect was discovered in this analysis as the two-way effect of video traffic by securing the audio stream was found to be significant  $(F_{(1,34)} = 4.55, p = .040)$  and is graphed in Figure 38. This new effect is likely related to the three-way effect of number of conversations by use of video by securing the audio stream. This effect is studied in greater detail by separating this data set into the constituent with- and without-AP components. Also, as can be expected from the top-level and with- and without-AP data, the one conversation data indicated that both the use of AP  $(F_{(1,34)} = 7.77, p = .007)$  and use of video  $(F_{(1,34)} = 51.49, p < .0005)$  is significant.

Also, the securing the audio stream effect is not significant ( $F_{(1,34)} = 0.91, p = .347$ ). The other two-way effects and the three-way effects were not significant as noted in Appendix D.

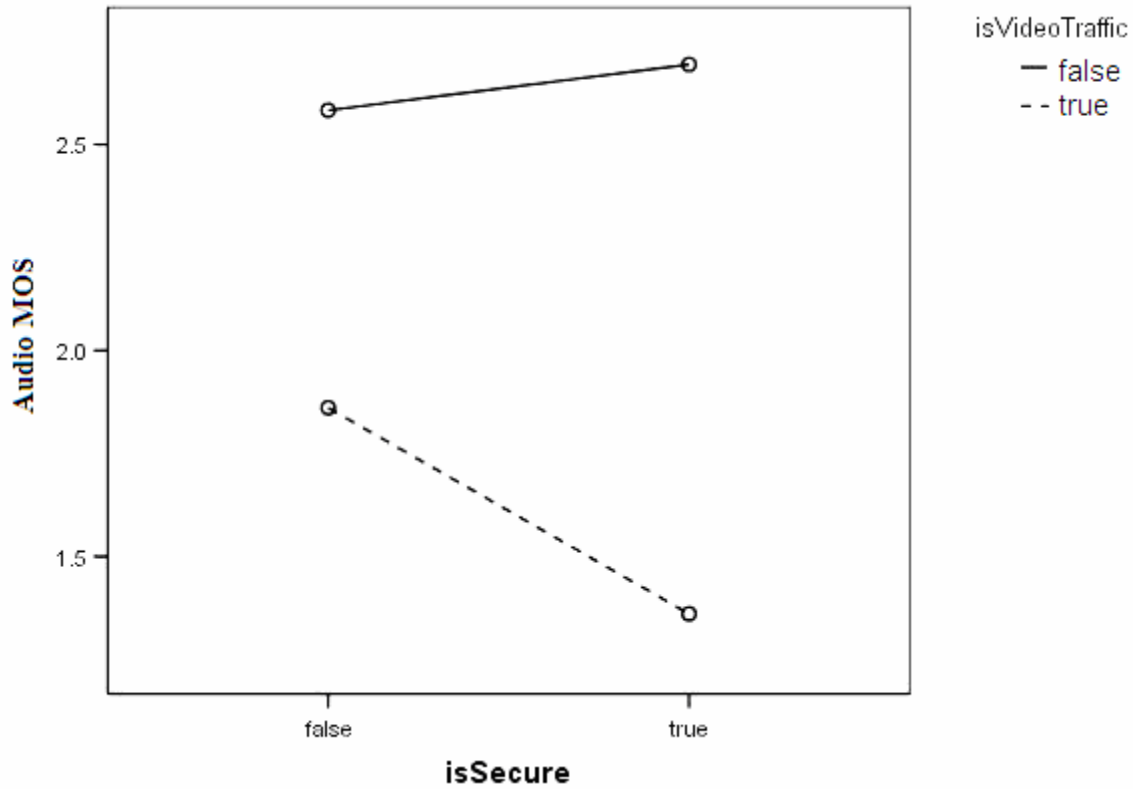


Figure 38. 2-way Effect Between the Use of Video and Security During One Conversation

Since the two-way effect of video traffic by securing the audio stream is found to have a significant effect on the audio MOS when one conversation is occurring, it makes sense to look at both conditions of use of the AP to see if the same effect is still present in both cases. It must be stated that the one conversation data has consistent results in terms of the significant effect of the use of video, use of the AP and the absence of effect of the securing the audio stream.

The use of AP data for one conversation is consistent with the combined one conversation data (with and without use of AP) in terms of the two-way effect of the use of video by securing the audio stream ( $F_{(1,34)} = 4.57, p = .040$ ) and is graphed in the Figure 39.

Additionally, the factor of video traffic is significant ( $F_{(1,34)} = 30.89, p < .0005$ ), whereas the securing the audio stream factor is not significant ( $F_{(1,34)} = 0.569, p = .456$ ).

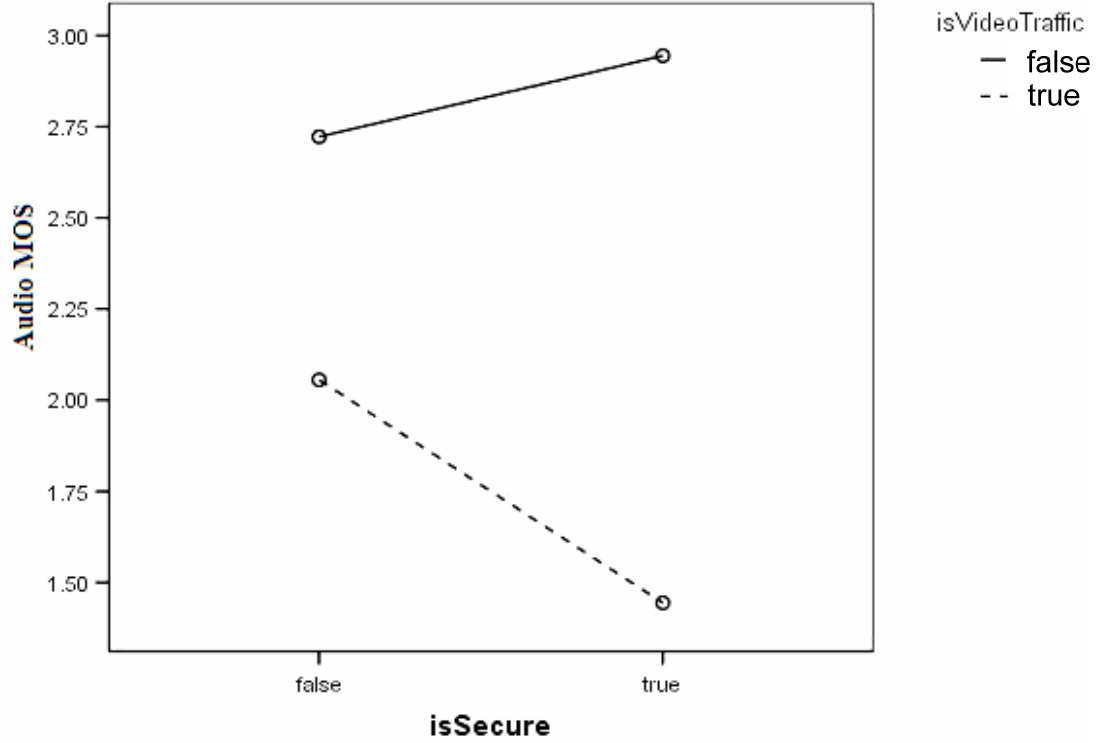


Figure 39. 2-way Effect between the Use of Video and Security (One Conversation with AP)

In a similar manner, the effect on the Audio MOS is studied when the AP is not used and only one conversation is occurring. In contrast to the combined and use of AP data, the two-way effect of use of video by the variable of securing the audio stream is not significant

( $F_{(1,34)} = 1.81, p = .188$ ). As expected, the video traffic is still significant

( $F_{(1,34)} = 45.17, p < .0005$ ) while securing the audio stream is not significant

( $F_{(1,34)} = .836, p = .367$ ).

It is difficult to explain the two-way effect of larger than expected degradation of the audio MOS when the audio was secure and video was present which was only significant when

the AP was being used. A possible reason why the two-way effect is only present in the AP terminal is that because the terminal has two more processes running than the non-AP network, the additional processing time of encryption and decryption of the audio RTP packet is enough to significantly reduce the quality of the audio stream. It is possible to test this explanation by running the multiple AP processes on multiple computers connected via Ethernet to the AP. By also dividing the one conversation data into the with- and without-video data sets and the with- and without-AP data sets, any inconsistencies or other patterns of the three-way effect may be determined.

All effects within the without-video one conversation set are consistent with the effects mentioned up to this point. The audio MOS analysis for the one conversation without video indicated a significant difference based on the use of the AP ( $F_{(1,34)} = 5.43, p = .026$ ). Both the two-way effect of use of AP by the factor of securing the audio stream ( $F_{(1,34)} = 0.443, p = .510$ ) and the factor of securing the audio stream ( $F_{(1,34)} = 0.150, p = .701$ ) are insignificant.

An unusual effect is first noted in the with-video one conversation data set which suggests that the securing the audio stream of the audio decreases the audio MOS. Otherwise all effects previously mentioned held in this data set, specifically, there is not a significant effect by the two-way factor use of AP by securing the audio stream ( $F_{(1,34)} = 0.663, p = .421$ ), and the use of AP ( $F_{(1,34)} = 4.15, p = .050$ ) and securing the audio stream ( $F_{(1,34)} = 5.97, p = .020$ ) are significant.

By now comparing the with- and without-video data of the one conversation data set it can be shown that securing the audio stream decreases the MOS only when the video is active, which is consistent with the analysis of the combined one-conversation data. This effect can possibly be explained as the effect of additional processing time required on a computer for the



encryption/decryption or in terms of increased network load or a combination of these two effects.

#### ***4.2.4 Analysis of the Two Conversation Audio MOS Data***

Referring back to the number of conversations effect on the audio MOS, the data related to when two conversations were occurring is next considered. As expected, the use of the AP ( $F_{(1,34)} = 10.49, p = .003$ ) and use of video ( $F_{(1,34)} = 28.72, p < .0005$ ) are both significant. However, all of the two-way and three-way effects were insignificant for the two conversation data as can be seen in Appendix D. The analysis of the between-subject effect of securing the audio stream yielded a result which is significant ( $F_{(1,34)} = 4.90, p < .034$ ).

When analyzing the data where two conversations were both routed through the AP, both the use of video ( $F_{(1,34)} = 16.346, p < .0005$ ) and securing the audio stream ( $F_{(1,34)} = 4.387, p = .044$ ) are significant. The two way factor of use of video by the factor of securing the audio stream is not significant ( $F_{(1,34)} = 2.62, p = .115$ ).

In contrast to the combined and with AP data, the without AP two-conversation data do not show a significant effect by the variable of securing the audio stream ( $F_{(1,34)} = 2.93, p = .096$ ). However, the use of video is significant ( $F_{(1,34)} = 6.24, p = .017$ ), while the two-way effect of use of video by securing the audio stream is not significant ( $F_{(1,34)} = 0.69, p = .411$ ).

A possible explanation of the significant effect of securing the audio stream degrading the audio MOS when there are two conversations and only when there is use of the AP can be explained by the experimental setup that has one station simulating all network traffic originating from terminals connected directly to the AP by Ethernet. When transmitting and receiving two

encrypted audio streams, it is possible that the audio MOS degrades because of the greater amount of processing that is required to encrypt and decrypt the audio signal. A follow-up study can determine if this is the cause of the effect by modifying the software to run multiple terminals simulating traffic routed through an AP.

#### 4.2.5 Analysis of Audio-Only Audio MOS Data

Further explanation of the six primary effects seen in the top level analysis can be made by discussing the audio-only and with-video data separately. Starting with the audio-only traffic, a data comparison is done between one conversation data with the within-subject variable of use of AP and the between-subject variable of securing the audio stream. Two of the within-subject factors are significant: the number of conversations ( $F_{(1,34)} = 20.93, p < .0005$ ) and use of the AP

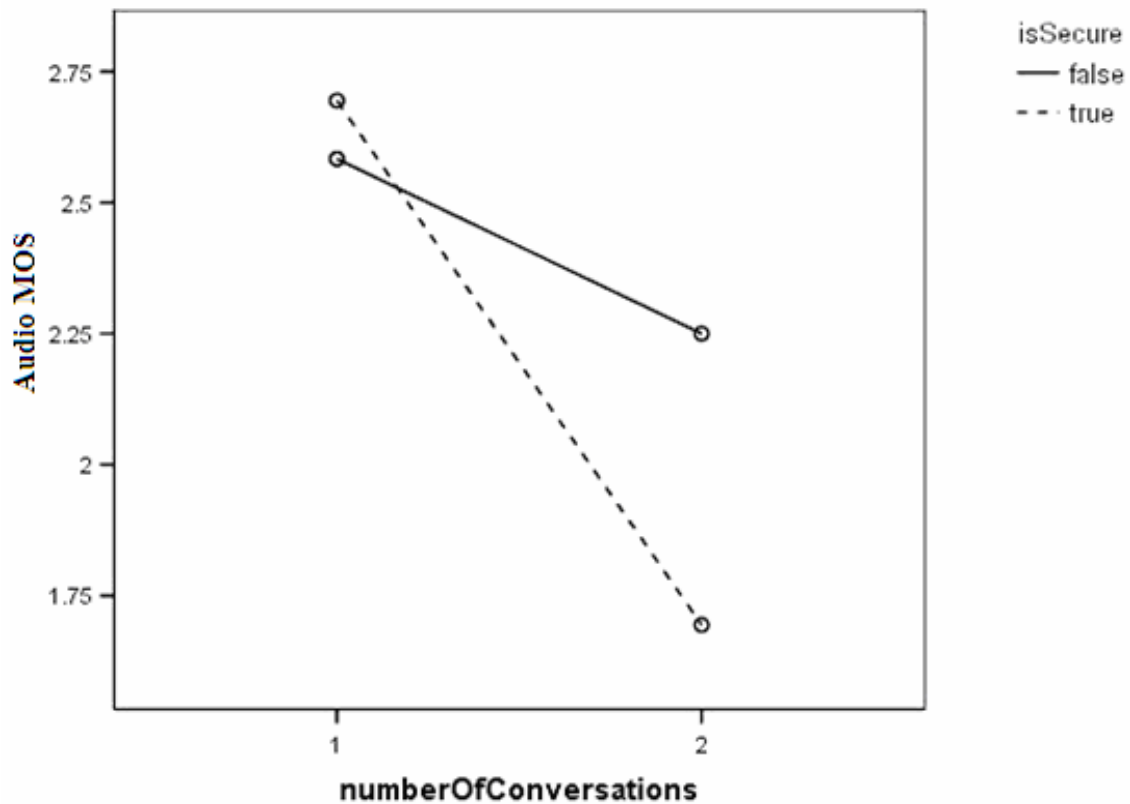


Figure 40. 2-way Effect Between the Number of Conversations and Security With Audio-Only

$(F_{(1,34)} = 9.01, p = .005)$ . Additionally, there exists a two-way effect: the number of conversations by securing the audio stream  $(F_{(1,34)} = 5.23, p = .029)$  which is graphed in Figure 40. However, the effect of securing the audio stream was insignificant in this part of the analysis  $(F_{(1,34)} = 1.12, p = .278)$ . Also, the two-way effect of securing the audio stream and use of the AP is not significant  $(F_{(1,34)} = 0.00, p = 1.000)$ .

When only audio data is considered with use of the AP, there exist a within-subject factor of number of conversations from one to four and the between-subject factor of securing the audio stream. The number of conversations is significant  $(F_{(3,102)} = 44.45, p < .0005)$  whereas the securing the audio stream was not significant  $(F_{(1,34)} = 3.85, p = .058)$ . The number of conversations by securing the audio stream two-way effect was not significant  $(F_{(3,102)} = 2.44, p = .069)$ .

Continuing the analysis of the audio data, only the non-AP data is considered in terms of the within-subject factor of number of conversations and the between subject factor of securing the audio stream. Only the number of conversations factor is significant  $(F_{(1,34)} = 18.00, p < .0005)$ , since the securing the audio stream factor is not significant  $(F_{(1,34)} = 9.94, p = .340)$ , along with the two-way effect of number of conversations by the factor of securing the audio stream  $(F_{(1,34)} = 2.00, p = .166)$ .

It is possible that the insignificance of the two-way effect of number of conversations by securing the audio stream effect is caused by the security reducing the audio MOS more severely in the two conversation data set than is expected without encryption. This would be caused by either greater demands on the computer for encryption and decryption and possibly by network congestion.

#### ***4.2.6 Analysis of Video-Enabled Audio MOS Data***

The video MOS data analysis procedure is very similar to the analysis of the audio MOS data. Analysis of the video MOS data suggests the number of conversations has an insignificant effect on the video MOS ( $F_{(1,34)} = 0.875, p = .356$ ). Also, the use of the AP ( $F_{(1,35)} = 3.95, p = .055$ ) is insignificant, in addition to all the two- and three-way effects (see Appendix D). However, the between-subjects variable of securing the audio stream is significant ( $F_{(1,34)} = 6.26, p = .017$ ).

The insignificance of the number of conversations must be addressed because of the uniqueness of this lack of effect. A possible explanation of the insignificance of this effect is that the network is already at capacity under the one conversation combined audio-video scenario. This is reasonable because the audio MOS  $1.75 \pm 0.30$  is already at a very low value. Therefore, it would be expected that a significant effect of adding additional traffic to the network would not significantly decrease the audio MOS.

Additionally, the significance of the securing the audio stream factor stands in contrast to other results. The data is suggesting that the additional processing time required for encrypting and decrypting the audio packets, coupled with sending and receiving of video, significantly decreases the audio MOS. However, this effect does not hold when comparing with- and without- AP data. This suggests that this effect is not significant, rather, it is related to whether the audio stream is routed through the AP.

It should be noted that the use of the AP is nearly significant. Since the test of significance of the between-subjects effect test nearly passed, it is of good practice to check the results of the with- and without-AP data for consistency with the combined data set.

In order to further analyze the insignificance of the number of conversations under the combined video and audio network conditions, the use of AP data is examined separately while

still considering the number of conversations factor. Data including use of AP meeting the above criteria do not suggest a significant effect by either the number of conversations

$(F_{(1,34)} = 0.88, p = .356)$  or securing the audio stream  $(F_{(1,34)} = 3.54, p = .069)$ .

Nearly identical results in terms of factor significance are obtained by considering only the non-AP data with video and audio traffic. No significance was found for either the number of conversations  $(F_{(1,34)} = 0.52, p = .476)$  or securing the audio stream  $(F_{(1,34)} = 3.13, p = .086)$ .

Since the individual data (with and without the AP) that had audio and video on the network do not suggest a significant effect of the securing the audio stream, the validity of the significance found for securing the audio stream factor in the combined (with and without AP) results can be brought into question. Also, the data analysis for the audio MOS with the combined audio and video on the secure wireless network suggests that the number of conversations is not significant consistently across all three data sets, which suggests that the network is already at capacity under the one conversation combined audio-video scenario. Therefore, additional traffic can not significantly decrease the audio MOS any further. This effect is the seventh consistent pattern in the data set.

#### **4.3 Analysis of Video MOS Data**

To help validate the audio MOS results in the combined audio and video data sets, the video data is next examined to determine if the number of conversations or the securing the audio stream of the audio stream has a significant effect on the video MOS. In contrast to the audio data results when both audio and video was on the network, the video MOS results are significantly affected by the number of conversations  $(F_{(1,34)} = 5.76, p = .022)$ . Additionally, securing the audio stream is a significant factor  $(F_{(1,34)} = 7.83, p = .008)$ . It should also be noted

that use of the AP is not a significant factor ( $F_{(1,34)} = 1.95, p = .172$ ), and there exist no significant two- or three-way effects (see Appendix D).

To help explain why the number of conversations was significant for the video MOS when it is not significant for the audio MOS under the same network conditions, it is hypothesized that subjects have a higher tolerance for delays in video which caused the video MOS to have a high contrast in scores between one conversation and two. Evidence to support this explanation includes a video MOS mean of  $1.972 \pm 0.151$  which is significantly higher than the audio MOS mean when video is also present on the network  $1.549 \pm 0.141$ , both of these values are calculated with a 95% confidence interval. The two mean values are the estimated marginal means that take into effect the number of conversations and whether the audio RTP is secured.

Another unique pattern seen within this data set is the insignificance of the effect of the AP on the video MOS. A possible way to reword this effect is that the efficiency of the stream originating from the AP makes a significant improvement to the audio traffic, and not the video traffic. An explanation of why the video traffic is not significantly improved by presence of the AP is that video RTP packets are above a critical size in which a significant efficiency is gained through use of the AP.

As noted in the audio MOS for the same data set, the securing the audio stream is a significant factor, which may help suggest that security processing delays are significantly impacting the video MOS. To help understand this effect further, the with- and without-AP data is looked at separately, to see if the same pattern seen in the audio MOS data helps explain this unique effect.

In order to further analyze the insignificance of the number of conversations under the combined video and audio network conditions, the AP use data will be looked at separately while still considering the number of conversations factor. Data including use of AP meeting the above

criteria also does not suggest a significant effect by the number of conversations

$(F_{(1,34)} = 1.39, p = .086)$ . However, the audio securing the audio stream does significantly affect the video MOS  $(F_{(1,34)} = 7.02, p = .012)$ .

Opposite results in terms of factor significance are obtained when considering only the non-AP video MOS data. The data suggests that the securing the audio stream is not significant  $(F_{(1,34)} = 3.38, p = .075)$ , and the number of conversations is significant  $(F_{(1,34)} = 5.65, p = .023)$ .

When analyzing just the one conversation data it is noted that both the use of AP  $(F_{(1,34)} = 0.42, p = .524)$  and securing the audio stream  $(F_{(1,34)} = 2.46, p = .126)$  are both insignificant.

When analyzing just the two conversation data, it is noted that the use of AP is not significant  $(F_{(1,34)} = 2.39, p = .132)$ , while the securing the audio stream is significant  $(F_{(1,34)} = 6.10, p = .019)$ .

Comparing all five of the video MOS data sets, two unique patterns are discovered that are not present in the audio MOS data. If there are two conversations or there is use of the AP, then if the audio is secured, then the video MOS value decreases. The use of AP with a secure audio channel can be expected to have a reduced video MOS score because a similar phenomenon is noticed with the audio MOS data since a combination of video and securing the audio stream did decrease audio MOS. When two conversations coupled with securing the audio stream decreases the video MOS, this indicates that there exists either a processing or network delay which causes the decreased the video MOS. Since the securing the audio stream processing was spread among the four terminals, the securing the audio stream must have caused a network delay only in the case of video MOS with two conversations.

The second pattern in the video MOS data set is that the video MOS was not affected by an increase in conversations from one to two in the use of AP data set. This pattern is likely

caused by a more efficient use of network resources by the AP as opposed to multiple terminals on a WLAN. This pattern can be thought of as similar to the used of the AP increasing the audio MOS as seen in prior analysis.

#### 4.4 Analysis of Intelligibility Data

For the purpose of increasing the validity of the MOS analysis and in the search for additional trends in secure wireless multimedia applications, data is collected on the number of correctly matched words transmitted with the audio stream. Based on the design of the experiment, two effects are tested in this part of the study; the effect of securing the voice real-time data as well as the effect of adding a video transmission simultaneous to the audio transmission. In contrast to the MOS data set, all the data sets for the intelligibility study have only one conversation occurring. The raw data collected consisted of the number correct in both the audio treatment and video treatment.

The data collected suggests that the effect of the video are significant ( $F_{(1,34)} = 7.07, p = .012$ ). However, both the effect of securing the audio stream ( $F_{(1,34)} = 0.69, p = .737$ ) and the two-way effect of use of video by securing the audio stream ( $F_{(1,34)} = 7.07, p = .409$ ) are insignificant.

These results are consistent with the MOS data results in terms of the lack of an effect by securing the audio stream as well as the presence of a significant effect by the use of video. The audio-only MOS value is  $2.83 \pm 0.36$  whereas the with-video audio MOS is  $1.75 \pm 0.30$ . Even with this relatively low value of the with-video audio MOS, the number of correctly matched words is  $18.1 \pm 0.54$ . This corresponds with a mean percent correct greater than 90% of the words spoken over the connection. In comparison, study of the G.723.1, G.728 and G.729 codecs, low packet loss percentages peak in the high 80% category [19].



## **4.5 Conclusion**

This chapter discusses the results and analysis of the subject data as prescribed in Chapter 3. Section 4.1 discusses how the analysis proceeded throughout the entire chapter. Section 4.2 discusses the results and analysis of the audio MOS data. Section 4.3 discusses the results of the video MOS data. Section 4.4 discusses the results and analyzes the intelligibility data.

## **V. Conclusion**

This chapter concludes the thesis. Section 5.1 summarizes the analysis completed in Chapter 4. Section 5.2 discusses recommendations for future research based on the results of this thesis. Section 5.3 discusses the relevance of this research effort.

### **5.1 Summary of Analysis**

A comprehensive survey of the questions and answers that are addressed in this study in addition to the primary effects observed in the results and analysis chapter is required because of the dispersal of the information within Chapter 4. To assist the reader, exceptions to the answers given in 5.1.1 are addressed in the secondary data effects subsection (5.1.2). Also, speculation regarding the technical reasons for these effects and answers is not addressed in this chapter as this information is contained within Chapter 4.

#### ***5.1.1 Study Questions and Answers***

The first research goal regards the impact of the number of conversations on the audio MOS. The answer to this question is that as the number of conversations increase from one to two, there is a decrease in the audio MOS by 23.5%. When the number of conversations increases from two to three, the audio MOS decreases by 36.6%. When the number of conversations increases from three to four, the audio MOS decrease by 9.8%.

The second research goal regards the impact of adding video capability to an audio conversation. It is observed that the addition of video decreases the audio MOS by 38.9%.

The third research goal regards the impact of the AP upon the audio MOS. The answer to this question is that when the AP forwards the audio traffic, the audio MOS increases by 18.4%.

The fourth research goal regards the effect of implementing a secure transmission of audio traffic upon the audio MOS. The answer to this question is that implementing secure audio has no significant effect on the audio MOS.

Next, these questions are addressed from the point of view of the video data. Note that question two is not applicable.

When considering question one which regards the impact of the number of conversations on the video MOS, it is determined that an increase in the number of conversation decreases the video MOS by 16.8%.

When considering question three which regards the impact of routing the traffic through an AP, it is determined (in contradiction to the audio data) that there is no significant effect on the video MOS.

When considering question four which regards the impact of securing the audio stream, it is determined that there is no significant effect on the video MOS.

### ***5.1.2 Secondary Data Effects***

With respect to audio MOS, increasing the number of conversations decreases an audio-only signal by 8.5% more than a combined voice and video signal. Also, with respect to audio MOS, in cases of two conversations with video, the use of a secured audio signal decreases the audio MOS by 12.3%. The final secondary data effect that is observed relative to audio MOS is that implementing secure audio decreases the audio MOS by an additional 12% when combined with the effect of increasing the number of conversations from one to two.

With respect to video MOS, if there are two conversations and the audio is secured, then the video MOS value will decrease by 23.3%. If an AP is used to forward traffic then if the audio is secured, then the video MOS value decreases by 19.5%. Also, the video MOS was not significantly affected by an increase in conversations from one to two when the AP was being used to forward the multimedia signal.

## **5.2 Recommendations for Future Research**

A full implementation of the SRTP protocol was not achieved as part of this thesis. However, many of the technical aspects of achieving the full SRTP status are documented in this thesis. Follow-up work can implement the remaining features of SRTP to include use of the AES instead of RSA, a method to securely transmit keys as detailed in the literature review as well as a block cipher mode of operation.

Additional details are needed in terms of the discrepancy between the analytical and computer-based models of wireless audio-video and the empirical results obtained within this study. Specifically in terms of analytical models, determining the answer to the question of whether it is possible to use a model based on a node's contribution to total network capacity to adequately describe the capacity of a wireless audio-video network is a useful question. In relationship to computer models, a recommendation of a series of changes to OPNET's wireless mode for the purpose of accurately predicting multimedia capacity of real networks would be very useful.

Of lesser importance, but still left unanswered in this thesis is if using one laptop to simulate all foreign WLAN traffic causes a confounding effect when trying to determine the effect of the AP as was noted in Chapter 4.

## **5.3 Relevance of the Current Investigation**

The use of voice and video-services that rely on IP-networks is becoming more common based on the widespread use of broadband connections that can support such traffic. Additionally, the rapid improvements seen within the WLAN hardware development community are making transmission of multimedia services more available to the wireless device user. When considering the next-generation 802.11n devices capable of transmission rates exceeding 100Mbps, it now becomes more likely that relatively high numbers of mobile devices can be

supported by wireless routers that are inexpensive enough for widespread deployment. Additionally, some mobile phones and devices can now support either cellular or IP-based technology. This thesis illustrates the limitations of the existing technology 802.11g technology, yet demonstrates that secure voice communications are available to a great number of Java application developers. Additionally, evidence within this thesis suggests that the suggested method of implementing secure voice transmission does not significantly affect the quality of the audio connection.

## **5.4 Conclusion**

This chapter concludes the thesis. Section 5.1 summarizes the analysis completed in Chapter 4. Section 5.2 discusses recommendations for future research based on the results of this thesis. Section 5.3 discusses the relevance of this research effort.

## **Appendix A. Informed Consent Letter**

### **Informed Consent Letter**

#### **For Research on Secure Wireless Multimedia**

You are invited to participate in a research study on secure wireless multimedia. This research is to be conducted by Lt Jason Seyba, USAF. This research is in partial fulfillment of a Master's degree program at the Air Force Institute of Technology (AFIT). The objective of this study is to determine ways to improve secure voice and video communication links over wireless LAN connections.

You will hear a series of voice messages over a communication network, in which you will evaluate your impression of the quality of the communication link by answering the following question about each stream:

Estimated Media Quality:

- 5. Excellent
- 4. Good
- 3. Fair
- 2. Poor
- 1. Bad

Additionally, you will be asked to match a spoken word with one of several choices that rhyme with the spoken word. Your individual answers will remain anonymous as you are not asked to, and should not; provide your name on the answer sheet. In addition, the data will not be grouped by organization. Therefore organizational answers are also kept anonymous.

Your participation is **COMPLETELY VOLUNTARY**. However, your input is important to improving analytical models for wireless secure multimedia capacities. You may withdraw from this study at any time without penalty, and your survey data will not be used in the research. Your decision to participate or withdraw will not jeopardize your relationship with your organization, the Air Force Institute of Technology, the Air Force, or the Department of Defense.

**PRIVACY ISSUES:** Records of my participation in this study may only be disclosed according to federal law, including the Federal Privacy Act, 5 U.S.C. 552a, and its implementing regulations (See Below).

If you have any questions about this request, please contact Lt Jason R Seyba - Phone (937) 656-4558; E-mail – jason.seyba@wpafb.af.mil

**HAVING READ THE INFORMATION PROVIDED, YOU MUST DECIDE WHETHER OR NOT TO PARTICIPATE IN THIS RESEARCH STUDY. YOUR SIGNATURE INDICATES YOUR WILLINGNESS TO PARTICIPATE.**

---

Participant's Signature      Date

---

Investigator's Signature      Date

### **Privacy Act Statement**

**Authority:** We are requesting disclosure of personal information, to include your Social Security Number. Researchers are authorized to collect personal information (including social security numbers) on research subjects under The Privacy Act-5 USC 552a, 10 USC 55, 10 USC 8013, 32 CFR 219, 45 CFR Part 46, and EO 9397, November 1943 (SSN).

**Purpose:** It is possible that latent risks or injuries inherent in this experiment will not be discovered until some time in the future. The purpose of collecting this information is to aid researchers in locating you at a future date if further disclosures are appropriate.

**Routine Uses:** Information (including name and SSN) may be furnished to Federal, State and local agencies for any uses published by the Air Force in the Federal Register, 52 FR 16431, to include, furtherance of the research involved with this study and to provide medical care.

**Disclosure:** Disclosure of the requested information is voluntary. No adverse action whatsoever will be taken against you, and no privilege will be denied you based on the fact you do not disclose this information. However, your participation in this study may be impacted by a refusal to provide this information.

## **Appendix B. Subject Information Sheet**

### **Information Sheet**

#### **For Research on Secure Wireless Multimedia**

You are invited to participate in a research study on secure wireless multimedia. This research is to be conducted by Lt Jason Seyba, USAF. This research is in partial fulfillment of a Master's degree program at the Air Force Institute of Technology (AFIT). The objective of this study is to determine ways to improve secure voice and video communication links over wireless LAN connections.

You will hear a series of voice messages over a communication network, in which you will evaluate your impression of the quality of the communication link by answering the following question about each stream:

Estimated Media Quality:

- 5. Excellent
- 4. Good
- 3. Fair
- 2. Poor
- 1. Bad

Additionally, you will be asked to match a spoken word with one of several choices that rhyme with the spoken word. Your individual answers will remain anonymous as you are not asked to provide your name to the software. In addition, the data will not be grouped by organization. Therefore organizational answers are also kept anonymous.

Your participation is COMPLETELY VOLUNTARY. However, your input is important to improving analytical models for wireless secure multimedia capacities. You may withdraw from this study at any time without penalty, and your survey data will not be used in the research. Your decision to participate or withdraw will not jeopardize your relationship with your organization, the Air Force Institute of Technology, the Air Force, or the Department of Defense.

**PRIVACY ISSUES:** Records of my participation in this study may only be disclosed according to federal law, including the Federal Privacy Act, 5 U.S.C. 552a, and its implementing regulations (See Below).

If you have any questions about this request, please contact Lt Jason R Seyba - Phone (937) 656-4558; E-mail – jason.seyba@wpafb.af.mil



### **Privacy Act Statement**

**Authority:** We are requesting disclosure of personal information, to include your Social Security Number. Researchers are authorized to collect personal information (including social security numbers) on research subjects under The Privacy Act-5 USC 552a, 10 USC 55, 10 USC 8013, 32 CFR 219, 45 CFR Part 46, and EO 9397, November 1943 (SSN).

**Purpose:** It is possible that latent risks or injuries inherent in this experiment will not be discovered until some time in the future. The purpose of collecting this information is to aid researchers in locating you at a future date if further disclosures are appropriate.

**Routine Uses:** Information (including name and SSN) may be furnished to Federal, State and local agencies for any uses published by the Air Force in the Federal Register, 52 FR 16431, to include, furtherance of the research involved with this study and to provide medical care.

**Disclosure:** Disclosure of the requested information is voluntary. No adverse action whatsoever will be taken against you, and no privilege will be denied you based on the fact you do not disclose this information. However, your participation in this study may be impacted by a refusal to provide this information.

## Appendix C. Secure Wireless Multimedia Raw Data

All of the tables in Appendix C provide the raw MOS scores related to the perceived quality of the audio or video stream. The subject numbers are listed in the first column for all tables. For the Audio-Only Raw MOS Data, the perceived quality of the audio stream in each of the six network conditions is available in the second through seventh column. As an example of how to read the Audio-Only Raw MOS Data, subject four rated network condition one 4/5 (Very Good).

Audio-Only Raw MOS Data						
Subject #	Network Condition					
	1	2	3	4	5	6
1	2	3	1	2	2	1
2	2	2	2	1	3	2
3	4	2	1	1	3	2
4	4	2	1	1	3	2
5	3	3	2	1	4	2
6	1	1	1	2	1	2
7	2	1	1	1	1	1
8	2	1	1	1	2	2
9	4	2	1	1	3	1
10	2	4	2	1	2	2
11	3	2	1	1	2	1
12	4	2	1	1	3	1
13	1	2	3	2	3	4
14	4	3	2	1	3	2
15	4	2	1	1	3	2
16	5	2	1	1	3	2
17	3	3	3	1	3	2
18	2	3	1	1	1	1
19	4	2	1	1	3	2
20	2	1	1	1	3	1
21	3	3	1	1	2	4
22	3	3	1	1	2	2
23	2	1	1	1	1	1
24	2	2	1	1	3	2
25	3	2	1	1	1	1
26	4	2	1	2	4	3
27	3	3	1	1	4	2
28	2	2	1	1	2	2
29	4	2	1	1	2	1
30	3	1	1	1	3	1
31	4	2	1	1	2	1
32	1	1	1	1	1	1
33	4	4	1	1	3	3
34	1	2	1	1	2	2
35	3	2	2	1	3	1
36	2	3	1	2	2	2

For the video-enabled audio and video raw MOS data, the perceived quality of the six network conditions is available in the second through seventh column. As an example of how to read the Video-Enabled Audio and Video Raw MOS Data, subject thirteen rated the first network condition's perceived audio quality 2/5 (Poor), whereas the perceived video quality is 3/5 (Fair).

Video-Enabled Audio and Video Raw MOS Data

Subject #	Network Condition (Audio/Video)							
	1 (A)	1 (V)	2 (A)	2 (V)	5 (A)	5 (V)	6 (A)	6 (V)
1	2	3	1	2	2	3	1	1
2	3	2	2	2	2	2	1	1
3	2	2	3	2	1	1	1	1
4	1	3	1	2	1	1	2	2
5	3	3	1	2	3	3	1	2
6	1	3	1	2	2	3	2	3
7	1	2	1	1	1	2	1	1
8	2	3	2	2	2	3	1	1
9	1	3	2	3	1	2	1	1
10	2	1	3	3	1	1	1	2
11	3	3	1	1	1	3	1	1
12	1	1	1	1	1	2	2	3
13	2	3	2	2	2	2	1	1
14	3	3	1	2	2	3	1	1
15	2	1	2	2	1	1	1	1
16	1	1	2	3	1	1	2	3
17	3	2	1	2	3	2	1	2
18	1	2	1	2	1	2	2	2
19	2	3	1	2	2	3	1	1
20	1	2	1	1	1	2	1	1
21	3	2	3	3	1	4	1	2
22	1	1	2	2	1	1	2	1
23	1	1	1	1	1	2	1	1
24	1	2	1	1	3	1	1	3
25	1	2	1	1	1	1	1	1
26	4	3	3	2	2	2	1	1
27	2	3	3	3	1	2	1	1
28	1	1	2	2	1	1	2	2
29	2	2	1	3	1	2	1	2
30	1	2	1	2	2	4	3	4
31	1	2	1	1	1	2	1	1
32	1	2	1	1	1	2	1	1
33	3	3	3	3	2	3	4	3
34	1	2	2	2	1	2	1	3
35	1	3	1	2	1	3	1	2
36	2	2	2	1	2	2	2	1

For the voice intelligibility data, the second column labeled “Audio” indicates the number of words that were selected correctly from a list of words that rhymed with the word that is spoken over an audio-only stream. The third column entitled “Video” indicates the number of words that were selected correctly when there existed both an audio and video stream.

Voice Intelligibility Raw Number Correct (Out of 20) Data

Subject #	Audio	Video
1	20	18
2	18	19
3	18	17
4	20	20
5	19	18
6	17	19
7	20	17
8	20	14
9	20	16
10	16	20
11	19	18
12	19	16
13	19	17
14	19	19
15	18	20
16	20	20
17	20	18
18	20	20
19	18	20
20	19	19
21	19	18
22	19	19
23	19	16
24	20	18
25	20	15
26	19	19
27	18	20
28	20	17
29	20	18
30	18	19
31	18	18
32	20	15
33	19	18
34	19	19
35	19	19
36	20	18

## Appendix D. Additional Subject Data Analysis

### Analysis of All Audio MOS Data

#### Within-Subjects Factors

Measure: audioMOS

numberOfConversations	isAPUsed	isVideoTraffic	Dependent Variable
1	FALSE	FALSE	A5
		TRUE	V5A
	TRUE	FALSE	A1
		TRUE	V1A
2	FALSE	FALSE	A6
		TRUE	V6A
	TRUE	FALSE	A2
		TRUE	V2A

#### Between-Subjects Factors

		Value Label	N
isSecure	0	FALSE	18
	1	TRUE	18

#### Tests of Between-Subjects Effects

Measure: audioMOS

Transformed Variable: Average

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Intercept	1,069.531	1	1,069.531	613.668	0.000
isSecure	5.837	1	5.837	3.349	0.076
Error	59.257	34	1.743		

### Tests of Within-Subjects Effects

Measure: audioMOS

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
numberOfConversations	11.281	1	11.281	14.335	0.001
numberOfConversations * isSecure	0.587	1	0.587	0.746	0.394
Error(numberOfConversations)	26.757	34	0.787		
isAPUsed	7.670	1	7.670	13.465	0.001
isAPUsed * isSecure	0.087	1	0.087	0.152	0.699
Error(isAPUsed)	19.368	34	0.570		
isVideoTraffic	41.253	1	41.253	99.545	0.000
isVideoTraffic * isSecure	0.281	1	0.281	0.679	0.416
Error(isVideoTraffic)	14.090	34	0.414		
numberOfConversations * isAPUsed	0.003	1	0.003	0.012	0.914
numberOfConversations * isAPUsed * isSecure	0.087	1	0.087	0.294	0.591
Error(numberOfConversations*isAPUsed)	10.035	34	0.295		
numberOfConversations * isVideoTraffic	5.281	1	5.281	8.514	0.006
numberOfConversations * isVideoTraffic * isSecure	4.253	1	4.253	6.857	0.013
Error(numberOfConversations*isVideoTraffic)	21.090	34	0.620		
isAPUsed * isVideoTraffic	0.281	1	0.281	0.420	0.521
isAPUsed * isVideoTraffic * isSecure	0.087	1	0.087	0.130	0.721
Error(isAPUsed*isVideoTraffic)	22.757	34	0.669		
numberOfConversations * isAPUsed * isVideoTraffic	0.003	1	0.003	0.018	0.895
numberOfConversations * isAPUsed * isVideoTraffic * isSecure	0.420	1	0.420	2.132	0.153
Error(numberOfConversations*isAPUsed*isVideoTraffic)	6.701	34	0.197		

## Analysis of One Conversation Audio MOS Data

### Within-Subjects Factors

Measure: audioMOS

isAPUsed	isVideoTraffic	Dependent Variable
FALSE	FALSE	A5
	TRUE	V5A
TRUE	FALSE	A1
	TRUE	V1A

### Between-Subjects Factors

		Value Label	N
isSecure	0	FALSE	18
	1	TRUE	18

### Tests of Within-Subjects Effects

Measure: audioMOS

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
isAPUsed	4.000	1	4.000	7.771	0.009
isAPUsed * isSecure	0.000	1	0.000	0.000	1.000
Error(isAPUsed)	17.500	34	0.515		
isVideoTraffic	38.028	1	38.028	51.489	0.000
isVideoTraffic * isSecure	3.361	1	3.361	4.551	0.040
Error(isVideoTraffic)	25.111	34	0.739		
isAPUsed * isVideoTraffic	0.111	1	0.111	0.345	0.561
isAPUsed * isVideoTraffic * isSecure	0.444	1	0.444	1.381	0.248
Error(isAPUsed*isVideoTraffic)	10.944	34	0.322		

### Tests of Between-Subjects Effects

Measure: audioMOS

Transformed Variable: Average

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Intercept	650.250	1	650.250	434.447	0.000
isSecure	1.361	1	1.361	0.909	0.347
Error	50.889	34	1.497		

## Analysis of Two Conversation Audio MOS Data

### Within-Subjects Factors

Measure: audioMOS

isAPUsed	isVideoTraffic	Dependent Variable
FALSE	FALSE	A6
	TRUE	V6A
TRUE	FALSE	A2
	TRUE	V2A

### Between-Subjects Factors

		Value Label	N
isSecure	0	FALSE	18
	1	TRUE	18

### Tests of Within-Subjects Effects

Measure: audioMOS

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
isAPUsed	3.674	1	3.674	10.494	0.003
isAPUsed * isSecure	0.174	1	0.174	0.496	0.486
Error(isAPUsed)	11.903	34	0.350		
isVideoTraffic	8.507	1	8.507	28.724	0.000
isVideoTraffic * isSecure	1.174	1	1.174	3.963	0.055
Error(isVideoTraffic)	10.069	34	0.296		
isAPUsed * isVideoTraffic	0.174	1	0.174	0.319	0.576
isAPUsed * isVideoTraffic * isSecure	0.063	1	0.063	0.115	0.737
Error(isAPUsed*isVideoTraffic)	18.514	34	0.545		

### Tests of Between-Subjects Effects

Measure: audioMOS

Transformed Variable: Average

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Intercept	430.563	1	430.563	416.772	0.000
isSecure	5.063	1	5.063	4.900	0.034
Error	35.125	34	1.033		



## Analysis of Video-Enabled Audio MOS Data

### Within-Subjects Factors

Measure: audioMOS

numberOfConversations	isAPUsed	Dependent Variable
1	FALSE	V5A
	TRUE	V1A
2	FALSE	V6A
	TRUE	V2A

### Between-Subjects Factors

	Value Label	N
isSecure	FALSE	18
	TRUE	18

### Tests of Within-Subjects Effects

Measure: audioMOS

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
numberOfConversations	0.563	1	0.563	0.875	0.356
numberOfConversations * isSecure	0.840	1	0.840	1.308	0.261
Error(numberOfConversations)	21.847	34	0.643		
isAPUsed	2.507	1	2.507	3.952	0.055
isAPUsed * isSecure	0.174	1	0.174	0.274	0.604
Error(isAPUsed)	21.569	34	0.634		
numberOfConversations * isAPUsed	0.007	1	0.007	0.038	0.846
numberOfConversations * isAPUsed * isSecure	0.063	1	0.063	0.344	0.562
Error(numberOfConversations*isAPUsed)	6.181	34	0.182		

### Tests of Between-Subjects Effects

Measure: audioMOS

Transformed Variable: Average

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Intercept	345.340	1	345.340	498.169	0.000
isSecure	4.340	1	4.340	6.261	0.017
Error	23.569	34	0.693		

## Analysis of Video MOS Data

### Within-Subjects Factors

Measure: audioMOS

numberOfConversations	isAPUsed	Dependent Variable
FALSE	FALSE	V5V
	TRUE	V1V
TRUE	FALSE	V6V
	TRUE	V2V

### Between-Subjects Factors

	Value Label	N
isSecure	false	18
	true	18

### Tests of Within-Subjects Effects

Measure: audioMOS

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
numberOfConversations	4.694	1	4.694	5.763	0.022
numberOfConversations * isSecure	0.111	1	0.111	0.136	0.714
Error(numberOfConversations)	27.694	34	0.815		
isAPUsed	1.000	1	1.000	1.946	0.172
isAPUsed * isSecure	0.028	1	0.028	0.054	0.818
Error(isAPUsed)	17.472	34	0.514		
numberOfConversations * isAPUsed	0.250	1	0.250	0.965	0.333
numberOfConversations * isAPUsed * isSecure	0.444	1	0.444	1.716	0.199
Error(numberOfConversations*isAPUsed)	8.806	34	0.259		

### Tests of Between-Subjects Effects

Measure: audioMOS

Transformed Variable: Average

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Intercept	560.111	1	560.111	701.715	0.000
isSecure	6.250	1	6.250	7.830	0.008
Error	27.139	34	0.798		

## Analysis of Intelligibility Data

### Within-Subjects Factors

Measure: audioMOS

isVideoUsed	Dependent Variable
FALSE	audioCorrect
TRUE	videoCorrect

### Between-Subjects Factors

	Value Label	N
isSecure	FALSE	18
	TRUE	18

### Tests of Within-Subjects Effects

Measure: audioMOS

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
isVideoUsed	17.014	1	17.014	7.071	0.012
isVideoUsed * isSecure	1.681	1	1.681	0.698	0.409
Error(isVideoUsed)	81.806	34	2.406		

### Tests of Between-Subjects Effects

Measure: audioMOS

Transformed Variable: Average

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Intercept	24,827.347	1	24,827.347	22,797.204	0.000
isSecure	0.125	1	0.125	0.115	0.737
Error	37.028	34	1.089		

## **Appendix E. Software Architecture**

This appendix discusses the architecture of the software that is used for the collection of data. Section E1 discusses the top layer of the software used for controlling the type and duration of the audio and video streams sent in the lower layer. Section E2 discusses the overall architecture of the lower layer used for sending and receiving secure audio and video traffic. Section E4 discusses the architecture of the transmission object and Section E5 discusses the receiving object architecture. Section E6 discusses how the connector object is secured. Section E7 describes the sequence of sending and receiving the secure object using the components described earlier in the appendix. Section E7 describes the method by which the traffic is encrypted and decrypted. Section E8 summarizes this appendix.

### **E1. RTP Traffic Control Layer**

One of the most critical requirements of the RTP traffic control layers is to be able to turn on and off audio transmissions based on user inputs to a graphical interface provided in the software. This is because the subjects indicate to the UI when the next audio/video clip should play and when the next matching word should be sent over the connection.

Two possibilities exist for controlling the RTP traffic, one is the use of well-developed open source server software; the Asterisk server was designed for such applications. However, several problems exist with integrating the Asterisk server software with a custom Java application hosted on a Windows LAN, specifically, Asterisk requires a Linux OS. Initial setup of the server was completed with difficulty, yet several additional complexities arose. To use the

Asterisk server for the experiment it is necessary to write the client side in compliance with SIP; this is a difficult and time-consuming task.

For this reason, the Asterisk-type capability is built from source. The first barrier encountered is the necessity to build a two-way connection over a single port so that the software could communicate simple messages to or from the server. All of the experiment software arose from FTP-type functionality that was originally developed from the Introduction to Computer Networks (CSCE 560) course taught at AFIT by Dr. Barry E. Mullins. This initial technology barrier is solved by building separate classes that extended the Java class thread named `ReadThread` and `ServerReadThread` for the client and server respectively. When both of the `ReadThread` classes are run, the method `read` is constantly called. Inside of the `read` method a `BufferedReader` object is used to receive a `String`. On the sending side a thread object is not required; instead a `DataOutputStream` object's `writeBytes` method is used to transmit the desired `String` object.

For the higher level capability, a UML diagram shown in Figure E1 is created for understanding the relationship between the classes `Avatar`, `AVServer`, `Collaborator` and `Team`. The `Avatar` class is designed to retain information specific to a client. Attributes include the

<b>MOS</b>	<b>Quality</b>	<b>Impairment</b>	<b>R-Value</b>
4.5	Excellent	Imperceptible	90
4.0	Very Good	Perceptible, but not annoying	80
3.5	Good	Very slightly annoying	70
3.0	Fair	Slightly annoying	60
2.5	Poor	Annoying	50
1.0	Bad	Very annoying	0

`InetAddress`, `ID`, `name`, the Boolean objects `isInSpeakSession` and `isBroadcasting`, a `Team` object and the controller class `AVClientController`. Additionally, objects related to the static and dynamic transmission of audio and video traffic have pointers in this class and are called:

soundTransmitter, videoTransmitter, staticAudioTransmitter and staticVideoTransmitter. All four of these objects descend from a class named AVTransmit which is discussed in Section E2. The purpose of each of these AVTransmit objects is to transmit real-time information to the team.

The AVServer Class holds such values as the objects serverInetAddress, and the integer serverPort as well as the controlling class AVClientController. All of the client application instances have an AVServer object that holds this information so that it can be used in the RTP Traffic Control Layer. The Team class has the attribute of an Avatar object representing the client instance, an integer for recording the current session the team is engaged in, as well as a series of HashMaps linking collaborator attributes to specific Collaborator objects associated with the team. A Collaborator object represents a user at a different terminal that can be referred to by name, port or id. Additional attributes of the Collaborator object include the Boolean objects of whether the Collaborator isInSpeakSession and two receivers of the class AVReceive are referred to as soundReceiver and videoReceiver. The AVReceive capability is described in Section E2. Also, it is logical to place UI component videoJPanel object of class JPanel in this layer, which is used by the videoReceiver object to display the real-time video on the user interface.

Following the development of the two-way connection over a single port solution and the model for the information that is transferred, a protocol simple, yet powerful enough to meet the requirements of the software could be designed. Since clients must be able to contact a server when starting the application, it is necessary for the client to have the server address and port set properly within the AVServer class. The clients also have the attributes of the Avatar set on application start. Thus, the information within both the AVServer and Avatar can be used to login to the system.

All of the basic steps for logging in from both the client and server side are shown in Figure E2. During the ClientController initialization procedure, the login method is called from within the ClientSignaller class. As an example of the contents of the login signal string, the

String object is: “login subjectA 134.120.71.129”. After the ClientSignaller sends this String using the sendSignal method, the ServerSignaller receives the signal and uses the already created Team object to create a collaborator with the attributes sent. Following this procedure, additional steps to include sending the id assigned back to the client and updating all other clients on the addition of a collaborator are completed.

The procedure to start a VoIP session is shown in Figure E3. After the AVClientController has been notified to start the session, most likely by a method call from within a visual component, the ClientSignaller is used to attempt the start of the session. After the ServerSignaller receives the request to start the session, the attemptStartAudio message is sent to the AVServerController to determine if either the sender or receiver is currently in a session that would eliminate the possibility of adding this new session. Assuming that no conflicts occur, the ServerSignaller is used by the AVServer Controller to confirm the start of the session. This signal is sent to both the sender of the request and the ClientSignaller of the intended receiver (marked as 2<sup>nd</sup> in the below diagram). Both the sender and the receiver ClientSignallers are called the attemptStartAudio method with the parameter of the Collaborator to start the session with. At this point the AVReceive object’s startAudio method is called the parameter of the Collaborator to start the session with. At this point the AVReceive object’s startAudio method is called.

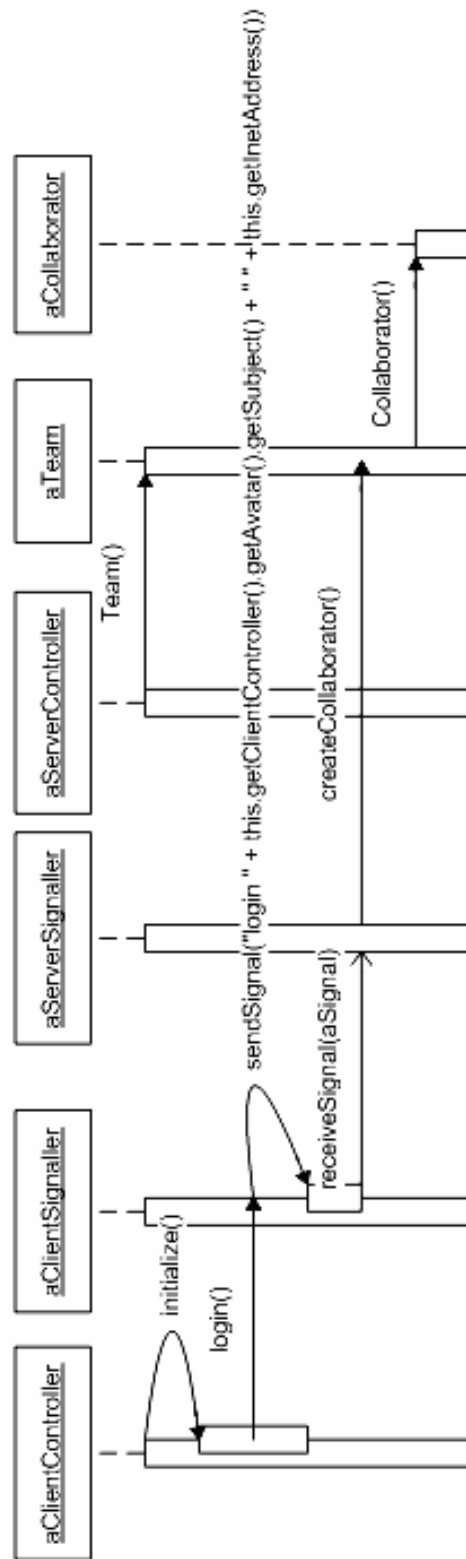


Figure E2. Construction of Collaborator Object

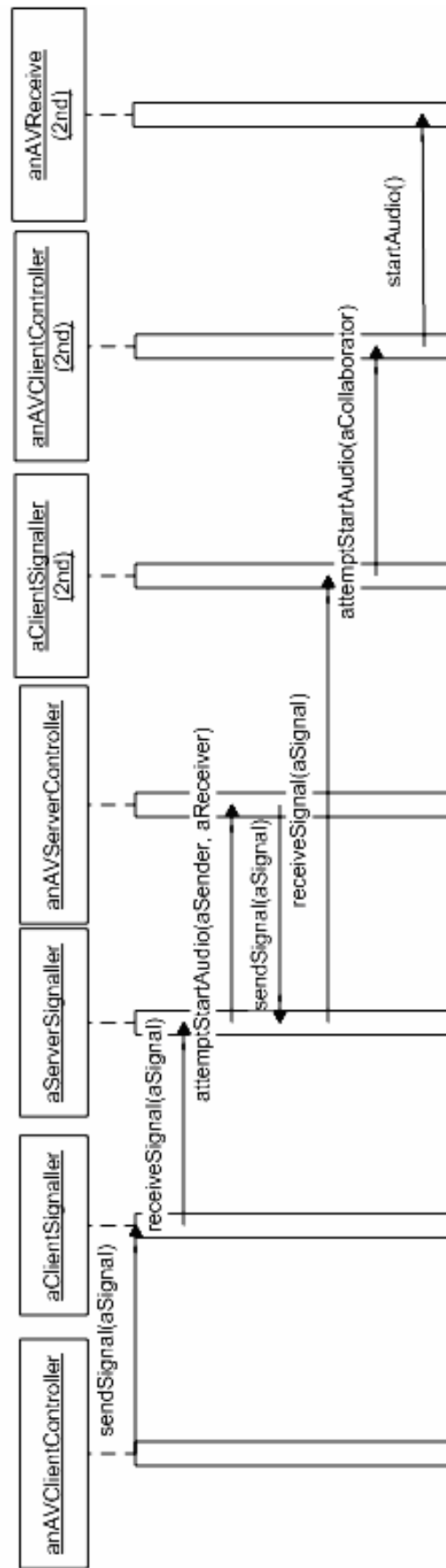


Figure E3. Sequence Diagram of start of Multimedia Session



## **E2. Secure RTP Traffic Layer**

There exist many methods to implement the capability of SRTP as described in Chapter 2 of this thesis. Publicly released code developed in C is widely available. However, since the sponsor's requirement of integrating the capability with a Java application was still needed, the only way to make use of the C would be to use the Java Native Interface library. This would allow the use of C source files, specifically the libSRTP software developed by Cisco Systems. However, even if the integration of the audio capability is successful, it cannot be assumed that the video content can be integrated within the Java application.

Another possibility is to alter the source code within the JMF (Java Media Framework) library. It is also possible to rewrite significant portions of the RTPSocket to allow for the encryption and decryption of packets. Additionally, there exist a well developed set of security libraries in the form of the Java Secure Socket Extensions to encrypt and decrypt the video or audio packet data. This solution, while complicated, still offered the best option from the point of view of development time.

## **E3. Secure AVTransmit Capability**

As part of the construction of the top-level AVTransmit object, three parameters must be passed; the creating object in addition to two other Boolean variables. The first of the Boolean objects is whether the object is used for audio transmissions; the second is whether this is a secure transmission. Upon initialization of this object, another internal class that has the full capability of the original Sun implementation of AVTransmit, in addition to the added security features is constructed. This original constructor method is passed the parameters of MediaLocator, InetAddress, Integer (for port to be used), Format, and the AVTransmitter. In actual implementation of this system, the MediaLocator is set to either the "jvasound://8000" for the

default sound input or “vfw://0” for the default video input. If a static audio or video file is to be transmitted, the file location is used in the MediaLocator constructor instead. The localhost address is used for the second parameter. An integer port base must be set by the AVConstructor prior to this initialization step, and is used as the third parameter. The format is set to null to indicate the default should be used.

Since the top-level AVTransmit object extends Thread, the start method is used to begin the transmission. Figure E4 illustrates the sequence to construct the AVTransmit object. Within this method the startTx method is called within the internal AVTransmit class. Two methods are called within the startTx method; createProcessor and createTransmitter. The createProcessor method creates and initializes the objects that supply the raw RTP data to the transmission object. The RTPManager object is created within the createTransmitter method. A major difference of the L3AV code compared with the default AVTransmit2 and AVReceive2 code is that a class that implements the RTPConnector interface is used for the initialization of the RTPManager. Before the AVSecureRTPSocket is initialized, the correct track of the DataSource is determined in addition to the correct port base, which is selected based on among other variables the current session of the team.

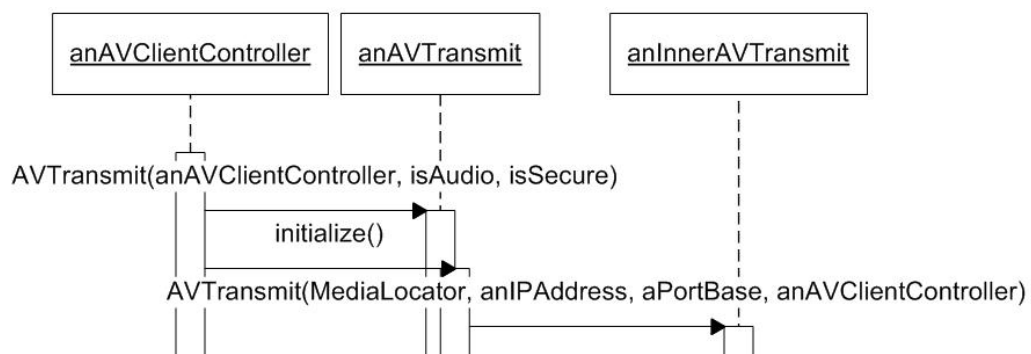


Figure E4. Sequence Diagram of AVTransmit Construction

Among the parameters of concern for the construction of the AVSecureRTPSocket are the address, port, the AVClientController, confirmation that this is a transmitting socket, the

AVSecureRTPSocket and a Boolean object that determines if secure features are being used. It should be stated that AVSecureRTPSocket is implementing the RTPSocket interface which consists of eleven methods. The entire capability of the AVSecureRTPSocket is explained in Section E6 because the receiving and transmitting capability must be explained simultaneously.

#### E4. Secure AVReceive Capability

For construction of the AVTransmit object, two parameters must be passed; a Collaborator object and a Boolean object to describe if the session is secure. Figure E5 illustrates the sequence to construct the AVReceive object. Upon initialization of the AVReceive object, an inner AVReceive class which is based largely on the SDN (Sun Developers' Network) AVReceive2 class. For this internal construction, the AVReceive object requires the parameters of a string for the collaborator's IP address, the integer values of the ports for audio and video, as well as a pointer to the JPanel used for displaying the video transmission.

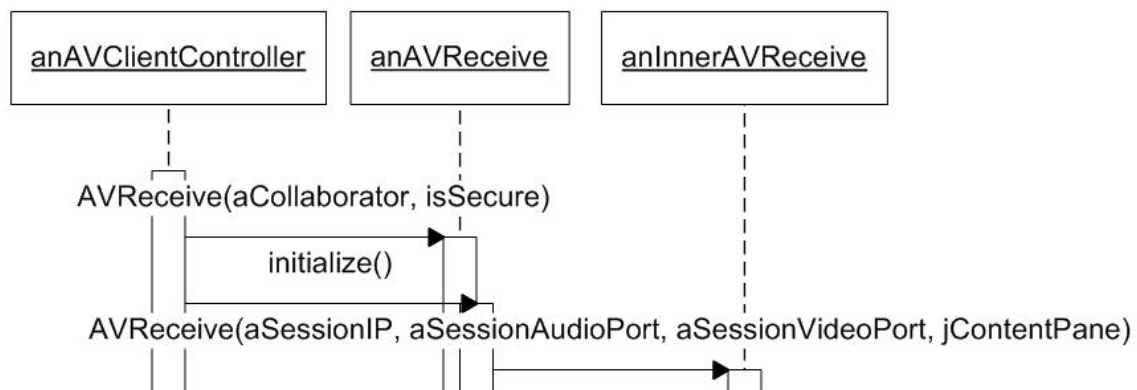


Figure E5. Sequence Diagram of AVReceive Construction

Also within the `initialize` method in Figure E5, a video or an audio session is created depending on the value of the Boolean `isAudio`. The two methods are `initializeAudioSession` and `initializeVideoSession`. Both of these methods use the same AVSecureRTPSocket constructor that is discussed in Section E3.

## E5. Secure Implementation of RTPConnector

Use of the RTPConnector interface allows the AVSecureRTPSocket to initialize the RTPManager class developed by Sun Microsystems. The use of a class diagram illustrated in Figure E6 eases the understanding of the implementation. The three static integers within the AVSecureRTPSocket are used to identify the three types of traffic used. The RTP and RTCP

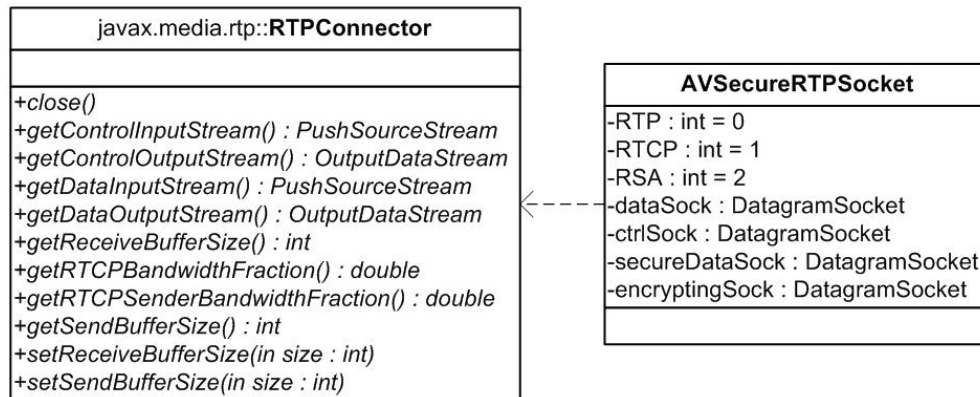


Figure E6. AVSecureRTPSocket Class Diagram

protocols have already been discussed in this paper, however, the RSA [39] (Rives Shamir Adlerman) algorithm has not. For implementation of the software, RSA encryption is used instead of AES as recommended in the SRTP protocol. The RSA algorithm is the famous security algorithm developed in 1977 by Ron Rives, Adi Shamir and Leonard Adlerman from MIT. The algorithm is more easily implemented within the JSSE (Java Secure Socket Extensions) software than is AES; the exact technique is described in detail in Section E7. Also, the reason four DatagramSocket objects are used within the AVSecureRTPSocket is described in

Section E6 which discusses the transmission and receipt of secure RTP traffic.

Understanding the purpose of the PushSourceStream and OutputDataStream is a prerequisite to understanding of the RTPConnector interface, therefore critical to the understanding of how JMF RTP traffic can be secured. The purpose of the PushSourceStream is to maintain a buffer and provide a stream of real-time audio or video data to a Java DatagramSocket using

DatagramPackets. In order for the entire capability of RTP to work, both a control and data stream must be provided to the RTPManager in the form of the two methods `getControlInputStream` and `getDataInputStream`. Only the `DataInputStream` has been secured in the L3AV implementation. Similarly, the purpose of the `OutputStream` is to maintain a buffer and receive from a `DatagramSocket` the `DatagramPackets` that contain the real-time audio or video data. Additionally, the `close` method closes all the objects of the classes

`PushSourceStream` and `OutputStream`. The `RTCP Bandwidth` and `SenderBandwidth` methods are used to notify the RTPManager of what frequency the RTCP messages should be sent. The last four variables: `getReceiveBufferSize`, `getSendBufferSize`, `setReceiveBufferSize` and `setSendBufferSize` are used to set up the two internal buffers. Additionally, the parameters listed in Table E1 are used to initialize the `AVSecureRTPSocket`.

**Table E1. AVSecureRTPSocket Constructor Parameters**

Parameter Name	Class
<code>addr</code>	<code>InetAddress</code>
<code>port</code>	<code>int</code>
<code>anAVClientController</code>	<code>AVClientController</code>
<code>isTransmitting</code>	<code>boolean</code>
<code>isSecure</code>	<code>boolean</code>

The `AVSecureRTPSocket` returns both a `PushSourceStream` and an `OutputStream` to the RTPManager object when the respective methods are called. The `AVSecureOutputStream` is illustrated in Figure E7. It can be seen that the class implements the method “write” from the `OutputStream` interface. The “data” parameter refers to the data that is supplied to the buffer, which is being maintained within the `AVSecureOutputStream`.

The offset and length integers enable the construction of a DatagramPacket from the contents of the buffer, so that the resulting packet can be supplied to the class DatagramSocket.

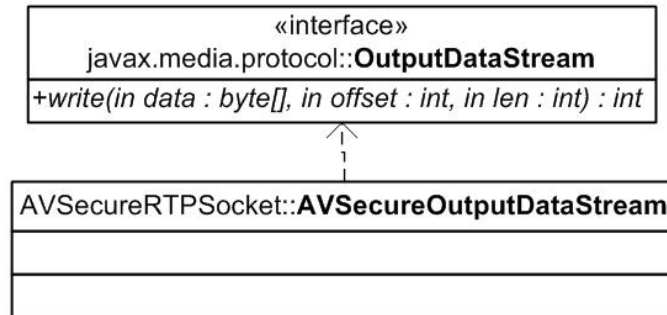


Figure E7. OutputStream Class Diagram

The complexity of the class AVSecurePushSourceStream is greater than the class OutputStream because reading from a data source and then supplying the contents of the data source requires the use of a thread. Also, since the capability of a thread is required, the three methods; start, kill and run were added. Since the class must implement the PushSourceStream interface, the read method returning an integer is required. The DatagramSocket uses this method with the parameters of a buffer, offset and length similar to the write method within the AVSecureOutputStream class. The minimum transfer size which is returned in the getMinimumTransferSize method is set to twice the value of the MTU (Maximum Transmission Unit) to ensure a balance between resource utilization and inherent delay within the system. The value is set at 2048 bytes because an 802.11 MTU of 1024 bytes generates greater access efficiency than a value of 512 bytes [31]. The SourceTransferHandler class purpose is to facilitate data transfers from the PushSourceStream to the DatagramSocket. The PushSourceStream contains the superinterfaces of SourceStream and Controls because five additional methods are required within the class AVSecurePushSourceStream. The method endOfStream always returns the value false because the AVSecurePushSourceStream returns a stream as long as the thread is alive. The ContentDescriptor is always returned null, because

descriptions of the content are handled in the upper layer in both the L3AV, as well as in H.323 and SIP implementations as opposed to within the RTP layer.

The content length is returned with the value unknown. The controls are returned with similar values for the same reason that the content descriptor values are absent. Thus, the `getControls` method returns an empty array and the `getControl` always returns a null object. The parameters required for the construction of the `AVSecurePushSourceStream` is shown in Table E2. The `DatagramSocket` indicates which socket receives the stream from the class `AVSecurePushSourceStream`. The address indicates which address the datagram packet will be sent from and the integer port identifies the correct originating process port. It is required that the `AVClientController` be passed because the L3AVSecurity features that handle the encryption and decryption of the datagram packets which have a pointer within the `AVClientController` object. The protocol can be one of the three protocols discussed above, RTP, RTCP or RSA. The

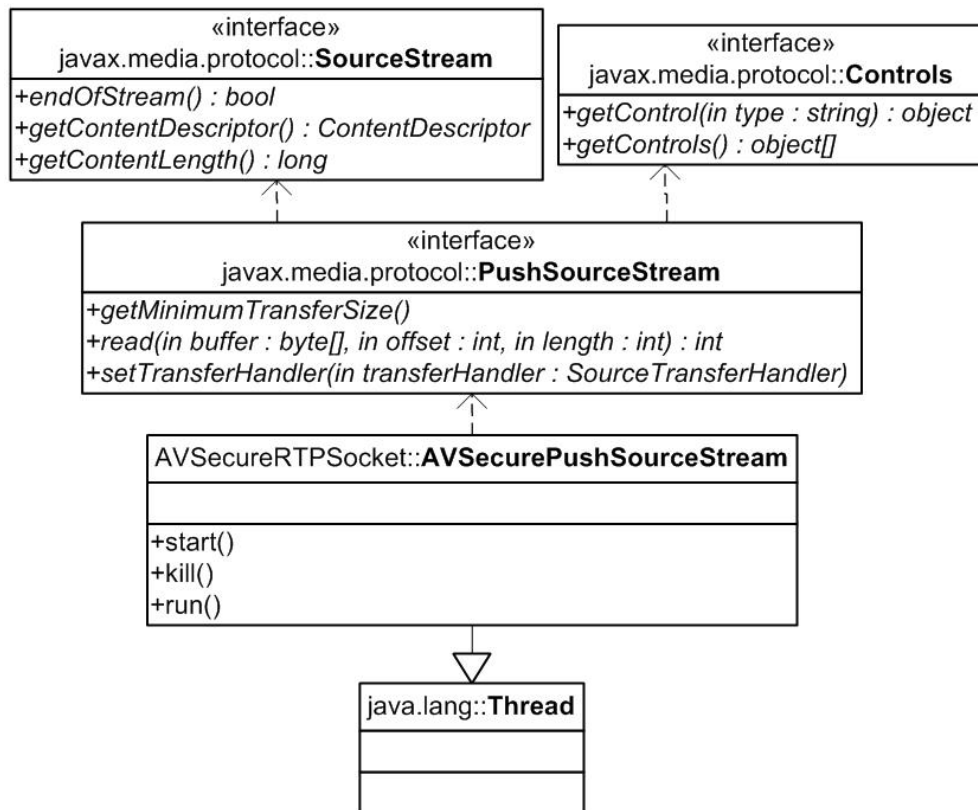


Figure E8. AVSecurePushSourceStream Class Diagram

receiverID is used for debugging purposes to identify which of the eight types of streams this AVSecurePushSourceStream object is pushing. Details of the eight types of streams are within Section E6. Additionally, the isShowingStreams parameter is also used for debugging purposes to view the contents of the data stream as received by the AVSecurePushSourceStream.

Table E2. AVSecurePushSourceStream Constructor

Parameter Name	Class
sock	DatagramSocket
addr	InetAddress
port	int
anAVClientController	AVClientController
aProtocol	int
aReceiverID	int
isShowingStreams	boolean

## E6. Transmission and Receipt of Secure RTP Traffic

With understanding of the primary components of and within the AVSecureRTPSocket complete, the method to secure the transmission between both the sender and receiver sockets can now be discussed. To allow basic understanding of the AVSecureRTPSocket, the case of no security is first considered. For this implementation, four streams are required since the AVSecureRTPSocket implements the RTPConnector interface. These four streams can be found in Table E3 as L3AV streams one, five, seven and eight. Stream one is referred to as “dataOutputStream” in the RTPConnector interface. For an insecure implementation of L3AV, the datagram socket type is data, whereas the encrypting datagram socket is used when the RTP traffic is secured. Continuing with the example of the insecure RTP traffic, stream six, referred to



as `dataInputStream` in the `RTPConnector` interface, is a datagram source which is used by the `RTPManager` to receive the data content. In a similar manner, streams seven and eight act as sink and source of the control data that is sent by the `RTPManager` class.

Table E3. L3AV Streams

L3AV Stream	Stream described in RTPConnector Interface	Sink or Source	Datagram Socket Type
1	Yes	Sink	encrypting or data
2	No	Source	encrypting
3	No	Sink	secureData
4	No	Source	secureData
5	No	Sink	data
6	Yes	Source	data
7	Yes	Sink	control
8	Yes	Source	control

When the L3AV capability secures the traffic, four additional streams are needed. The capability of the six streams associated with the encryption and decryption of data is illustrated in Figure E7. The encrypt datagram socket is associated with streams one and two which act as sink and source respectively. Stream one is responsible for accepting the incoming RTP traffic and then supplying the traffic to stream two which is responsible for supplying the traffic to a process named `AVSecurityRTPThread` which is described Section E7. Following the encryption process, streams three and four are used by the `AVSecurityRTPThread` class and the secure datagram socket to send and receive the secured RTP traffic between the two endpoints. Finally, streams five and six act as a sink from the `AVSecurityRTPThread` class and as a source to the `RTPManager` class on the receiving end.

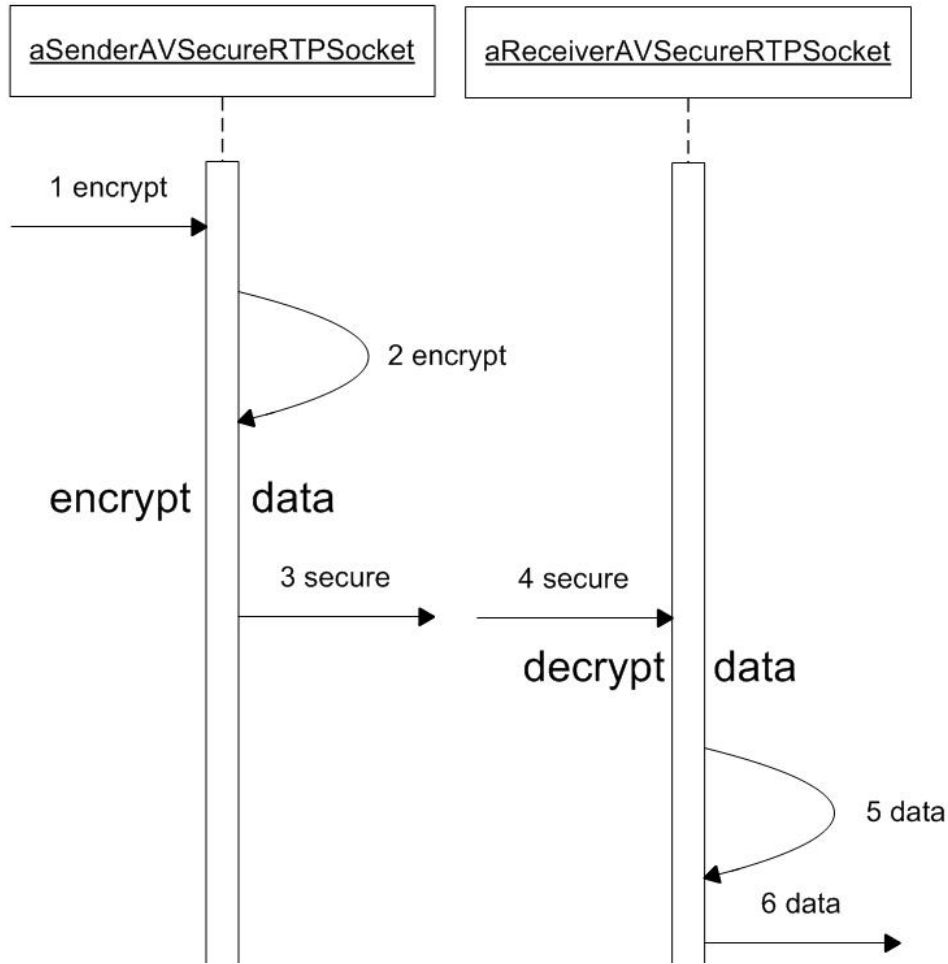


Figure E9. Streams Associated with Encryption and Decryption of Data

### E7. Encryption and Decryption Method

The four streams associated with encryption and decryption (streams 2-5 in Figure E9) are attributes of the class AVSecurityRTPThread (See Figure E10). The purpose of this thread is to continuously encrypt and decrypt RTP traffic in cooperation with the AVSecureRTPSocket and to also ensure the packets are sent to the correct destination. Additionally, two objects of the class AVSecureRTPPacketReceiver are used to receive the encrypted and unencrypted packets for the purpose of decryption and encryption.

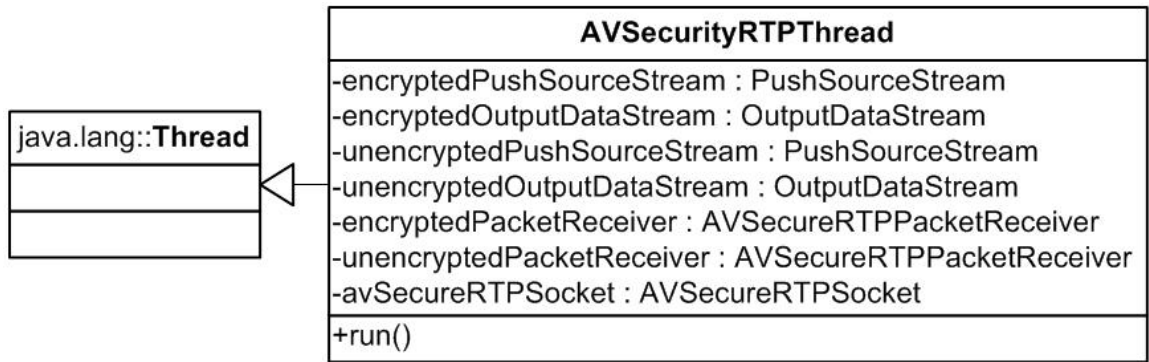


Figure E10. AVSecurityRTPThread Class Diagram

The class diagram for the AVSecureRTPPacketReceiver illustrated in Figure 11 extends from the class RTPPacketReceiver which contains the method receiveFrom. The receiveFrom method supplies a Packet which is either encrypted or decrypted as controlled by the AVSecurityRTPThread.

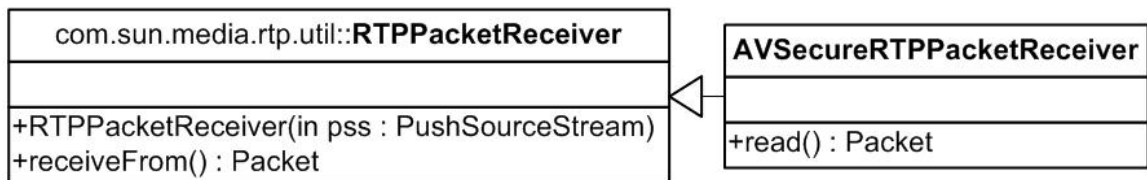


Figure E11. AVSecurityRTPPacketReceiver Class Diagram

Figure E12 describes how the unencrypted packet is encrypted. After the receipt of an unencrypted packet through the use of the read method within the unencrypted packet receiver, the datagram packet is encrypted through the use of the AVSecurity encryptAll method. It is possible to use any type of security algorithm within this method; however as stated before, the RSA algorithm is implemented. By adding AES encryption and decryption and a block cipher mode, the SRTP specification can be achieved. The addition of these features to L3AV is an excellent candidate for additional follow-on work as it an additional step towards fulfilling the SRTP specification and greatly increases the inherent security of the system.

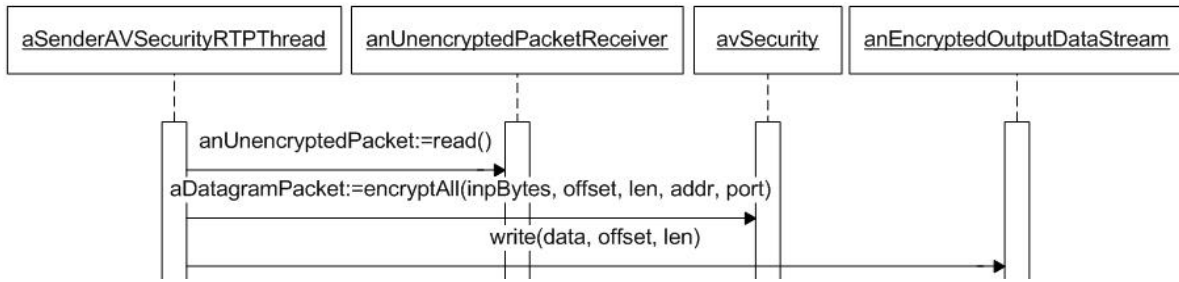


Figure E12. Encryption Sequence Diagram

Using nearly an identical approach as the encryption process, the decryption process uses the decryptAll method within the AVSecurity object to obtain an unencrypted datagram packet which can be sent to the unencrypted output data stream. Figure E13 illustrates the decryption procedure as implemented in the software.

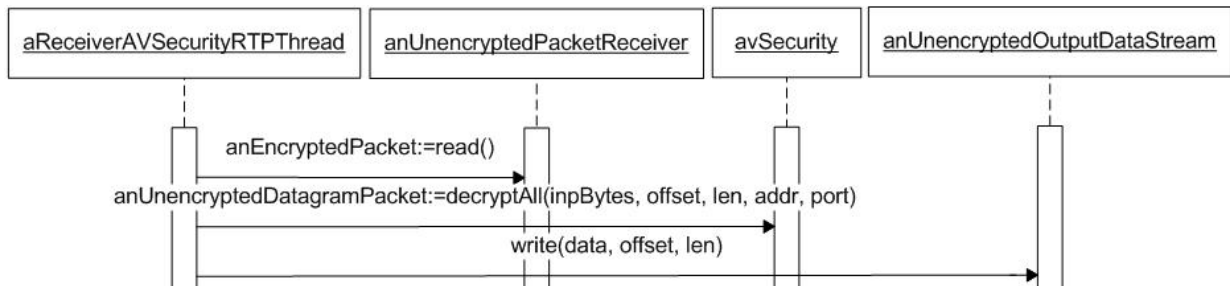


Figure E13. Decryption Sequence Diagram

## E8. Conclusion

This appendix describes the design of the experimental software. Section E1 explains how the RTP Traffic Control layer works to create the different network conditions that are to be studied. Section E2 explains the purpose of the RTP Traffic Layer. Section E3 describes how the transmission object works as part of the secure system. Section E4 describes how the receiving object works as part of the secure system. Section E5 describes the class of secure sockets that are used in this software implementation. Section E6 describes the transmission and receipt of secure traffic. Section E7 describes how the data contents of the RTP traffic is encrypted before transmission and receipt over a possibly compromised data channel.

## Bibliography

1. Advanced Encryption Standard <http://www.quadibloc.com/crypto/co040401.htm>
2. Advanced Encryption Standard. In Wikipedia, The Free Encyclopedia. Retrieved from [http://en.wikipedia.org/w/index.php?title=Advanced\\_Encryption\\_Standard&oldid=69715376](http://en.wikipedia.org/w/index.php?title=Advanced_Encryption_Standard&oldid=69715376) ( 2006)
3. Andeasen, Bauer, Wing, “Session Description Protocol (SDP) Security Descriptions for Media Streams”, RFC 4568 (2006)
4. Anderson, Norman H; Empirical Direction in Design and Analysis, Lawrence Erlbaum Associates (2001)
5. Arkko, Carrara, Lindholm, Naslund, Norrman, “MIKEY: Multimedia Internet KEYing”, RFC 3830 (2005)
6. Baodian, Liu, Dongsu, Ma Wenping and Wang Xinmei, "Property of finite fields and its cryptography application" *IEEE Electronics Letters Online No. 20030444* (2003)
7. Bellare, Mihir, Ran Canetti, Hugo Krawczyk, "Keying Hash Functions for Message Authentication" *Advances in Cryptology - Crypto 96 Proceedings*, Lecture Notes in Computer Science Vol. 1109, N. Koblitz ed., Springer-Verlag (1996)
8. Block cipher modes of operation. In Wikipedia, The Free Encyclopedia. Retrieved from [http://en.wikipedia.org/w/index.php?title=Block\\_cipher\\_modes\\_of\\_operation&oldid=68783487](http://en.wikipedia.org/w/index.php?title=Block_cipher_modes_of_operation&oldid=68783487) (2006)
9. Brace, Nicole, Richard Kemp, and Rosemary Snelgar. SPSS for Psychologists: A Guide to Data Analysis using SPSS for Windows. Mahwah, NJ: Lawrence Erlbaum Assoc, Inc. (2003)
10. Clark, A. D. “Modeling the Effects of Burst Packet Loss and Recency on Subjective Voice Quality”, Suwanee, GA *IP Telecommunications Conference* (2001)
11. Daemen, Joan, Vincent, Rijmen *Proposal Rijndael* <http://csrc.nist.gov/CryptoToolkit/aes/rijndael/> (1998)
12. Federal Information Processing Standards Publication 180-2 "Secure Hash Standard" <http://csrc.nist.gov/publications/fips/fips180-2/> (2002)

13. Garg, S. and M. Kappes, "An experimental Study of Throughput for UDP and VoIP Traffic in IEEE 802.11b Networks," *Wireless Communications and Networking Conference* (2003)
14. GL Communications, "ITU Algorithms" <http://www.gl.com/ITUalgorithms.html> (2005)
15. Hallivuori, Ville "Real-time Transport Security," *Helsinki University of Technology Telecommunications Software and Multimedia Laboratory Seminar on Network Security* (2000)
16. Hegde, N., A. Proutiere, J. Roberts, "Evaluating the voice capacity of 802.11 WLAN under distributed control," *IEEE Local Area Network Metropolitan Area Network Workshop* (2005)
17. Hole, D. and F. Tobagi "Capacity of an IEEE 802.11b Wireless LAN supporting VoIP," *IEEE International Communications Conference* (2004)
18. Home Audio Meter Hearing Test Software, <http://www.audiometer.co.uk/> (2007)
19. Hooper, J.B. and M.J. Russell. "Objective quality analysis of a Voice over Internet Protocol system" *Electronics Letters* 36:22 pp1900-1902 (2000)
20. IEEE 802.11e Project  
[http://grouper.ieee.org/groups/802/11/Reports/tge\\_update.htm](http://grouper.ieee.org/groups/802/11/Reports/tge_update.htm) (2006)
21. IEEE Std 802.11g <http://standards.ieee.org/getieee802/802.11g-2003.pdf> (2003)
22. IETF RFC 3261 <http://www.ietf.org/rfc/rfc3261.txt> (2002)
23. ITU Recommendation G.711 <http://www.itu.int/rec/T-REC-G.711-198811-I/en> (1988)
24. ITU Recommendation G.729 <http://www.itu.int/rec/T-REC-G.Imp729-200604-I/en> (2006)
25. ITU Recommendation H.323 <http://www.itu.int/rec/T-REC-H.323-200606-I/en> (2007)
26. Joint Video Team of ITU-T and ISO/IEC JTC 1, "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC)," document JVT-G050r1, May 2003; technical corrigendum 1 documents JVT-K050r1 (non-integrated form), (2004)

27. Li, B. and R. Battiti. "Performance Analysis of An Enhanced IEEE 802.11 Distributed Coordination Function Supporting Service Differentiation" Stockholm, Sweden, *Quality of Future Internet Services* (2003)
28. Man-in-the-middle attack. In Wikipedia, The Free Encyclopedia. from [http://en.wikipedia.org/w/index.php?title=Man-in-the-middle\\_attack&oldid=69805325](http://en.wikipedia.org/w/index.php?title=Man-in-the-middle_attack&oldid=69805325) (2006)
29. Maguire, G. Q. Jr., Lecture notes [maguire@it.kth.se](mailto:maguire@it.kth.se) (2006)
30. Mean Opinion Score. In Wikipedia, The Free Encyclopedia. from [http://en.wikipedia.org/wiki/Mean\\_Opinion\\_Score](http://en.wikipedia.org/wiki/Mean_Opinion_Score) (2006)
31. Nguyen, Thy T.T. Armitage, Grenville, "Quantitative Assessment of IP Service Quality in 802.11b and DOCSIS Networks" *Australian Telecommunications Networks and Applications Conference* (2004)
32. NIST / SEMATECH *e-Handbook of Statistical Methods* <http://www.itl.nist.gov/div898/handbook> (2007)
33. Porter, Thomas; Practical VoIP Security; Syngress (2006)
34. RFC 3015 - MEGACO. Media Gateway Control. <http://www.faqs.org/rfcs/rfc.3015.html> (2001)
35. RFC 3550 – Real Time Transport Protocol <http://www.rfc-editor.org/rfc/rfc3550> (2003)
36. RFC 3711 – The Secure Real-time Transport Protocol (SRTP) <http://www.ietf.org/rfc/rfc3711.txt> (2004)
37. RFC 4835 – <http://www.faqs.org/rfcs/rfc4835>
38. Rijmen and Oswald, "Update on SHA-1", *Lecture Notes in Computer Science* 3376, pp.~58-71 (2005)
39. RSA Algorithm <http://www.rsasecurity.com/rsalabs/node.asp?id=2248> (2007)
40. Rubino, Gerardo, Martin Varela and Jean-Marie Bonnin; Wireless VoIP at Home: Are We There Yet? *Measurement of Speech and Audio Quality in Networks* (2005)
41. Savard John J. The Advanced Encryption Standard (Rijndael). From <http://www.quadibloc.com/crypto/co040401.htm> (1998)

42. Schroeder, M. R. and B. S. Atal, "Code-excited linear prediction (CELP): high-quality speech at very low bit rates," in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 10, pp. 937-940, (1985)
43. Schulzrinne, H. and S. Casner, "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, (2003)
44. Seyba J., B. Mullins, G. Bonafede "Audio-Video Capacity of an IEEE 802.11g Wireless LAN" In W. W. Smari and W. McQuay (Eds.), International Symposium on Collaborative Technologies and Systems Orlando, FL: IEEE (2007)
45. Souza, J., L. Carvalho, E. Mota, R. Aguiar, A. Lima, A. Barreto. "An E-Model Implementation for Speech Quality Evaluation in VoIP Systems", Cartagena, Murcia, Spain, IEEE ISCC, pp. 933-938 (2005)
46. SPSS Inc. SPSS 15.0 Brief Guide. Chicago, <http://www.spss.com>, (2006)
47. Sullivan, G., P. Topiwala, and A. Luthra, "The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions," SPIE Conference on Application of Digital Image Processing XXVII. (2004)
48. TIPHON (Telecommunications and Internet Protocol Harmonization Over Networks) Working Group. "TIPHON Release 3; Technology Compliance Specification; Part 5: Quality of Service (QoS) Measurement Methodologies", *European Telecommunications Standards Institute* 101 329-5 v1.1.1 (2000)
49. Yuan, Quan Songping Li "A New Efficient ID-Based Authenticated Key Agreement Protocol "Cryptography ePrint Archive (2005)
50. Zimmerman, P., "ZRTP: Extensions to RTP for Diffie-Hellman Key Agreement for SRTP", RFC 3830 (2007)



## **Vita**

Jason R. Seyba, First Lieutenant, USAF graduated from Iowa State University in Ames, Iowa, with a Bachelor of Science in Mechanical Engineering in May 2003. In May 2004, he was commissioned as a Second Lieutenant in the United States Air Force through the Officer Training School, Maxwell Air Force Base, Montgomery, Alabama. Currently, he is the Lead Collaborative Logistics Engineer assigned to the Logistics Readiness Branch of the Air Force Research Laboratory, Wright-Patterson Air Force Base, Dayton, Ohio. In this assignment, he has held positions such as Program Manager of the Decision Support Technologies for Logistics Readiness Center, Technical Lead of the Virtual Space Logistics Readiness Center (VSLRC) Program and Manager of the Logistics Living Laboratory.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 074-0188	
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p><b>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</b></p>					
<b>1. REPORT DATE (DD-MM-YYYY)</b> 14-06-2007		<b>2. REPORT TYPE</b> Master's Thesis		<b>3. DATES COVERED (From – To)</b> Jun 2006 – May 2007	
<b>4. TITLE AND SUBTITLE</b>  Voice and Video Capacity of a Secure Wireless System				<b>5a. CONTRACT NUMBER</b>	
				<b>5b. GRANT NUMBER</b>	
				<b>5c. PROGRAM ELEMENT NUMBER</b>	
<b>6. AUTHOR(S)</b>  Seyba, Jason, R., 1 <sup>st</sup> Lieutenant, USAF				<b>5d. PROJECT NUMBER</b>	
				<b>5e. TASK NUMBER</b>	
				<b>5f. WORK UNIT NUMBER</b>	
<b>7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)</b> Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765				<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>  AFIT/GCS/ENG/07-14	
<b>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> AFRL/HEAL Attn: Mr. Paul D. Fass 2698 G Street WPAFB OH 45433-7604 DSN: 986-4390				<b>10. SPONSOR/MONITOR'S ACRONYM(S)</b>	
				<b>11. SPONSOR/MONITOR'S REPORT NUMBER(S)</b>	
<b>12. DISTRIBUTION/AVAILABILITY STATEMENT</b> APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.					
<b>13. SUPPLEMENTARY NOTES</b>					
<b>14. ABSTRACT</b> <p>Improving the security and availability of secure wireless multimedia systems is the purpose of this thesis. Specifically, this thesis answered research questions about the capacity of wireless multimedia systems and how three variables relate to this capacity. The effects of securing the voice signal, real-time traffic originating foreign to a wireless local area network and use of an audio-only signal compared with a combined signal were all studied. The research questions were answered through a comprehensive literature review in addition to an experiment which had thirty-six subjects using a secure wireless multimedia system which was developed as part of this thesis effort. Additionally, questions related to the techniques for deploying wireless multimedia system including the maturity and security of the technology were answered. The research identified weaknesses in existing analytical and computer models and the need for a concise and realistic model of wireless multimedia systems. The culmination of this effort was the integration of an audio-video system with an existing research platform which is actively collecting data for the Logistics Readiness Branch of the Air Force Research Laboratory.</p>					
<b>15. SUBJECT TERMS</b> <p>VoIP, Voice over IP, Video over IP, Secure Real Time Protocol, Wireless Multimedia, Human Subjects, Mean Opinion Score, Intelligibility</p>					
<b>16. SECURITY CLASSIFICATION OF:</b>			<b>17. LIMITATION OF ABSTRACT</b>  UU	<b>18. NUMBER OF PAGES</b>  102	<b>19a. NAME OF RESPONSIBLE PERSON</b> Barry E. Mullins, Ph.D. (ENG)
<b>REPORT</b> U	<b>ABSTRACT</b> U	<b>c. THIS PAGE</b> U			<b>19b. TELEPHONE NUMBER (Include area code)</b> (937) 255-3636 x 7979 barry.mullins@afit.edu

**Standard Form 298 (Rev. 8-98)**

Prescribed by ANSI Std. Z39-18